University of Alberta

Library Release Form

Name of Author: Kwan Lai Ling

Title of Thesis: Visualization of ATM Network Data

Degree: Master of Science

Year this Degree Granted: 1998

University of Alberta

Visualization of ATM Network Data

by

Kwan Lai Ling ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of **Master of Science**.

Department of Computing Science

Edmonton, Alberta
Spring 1998

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Visualization of ATM Network Data** submitted by Kwan Lai Ling in partial fulfillment of the requirements for the degree of **Master of Science**.

Abstract

This thesis describes a 3D visualization system that is meant to assist engineers in analyzing ATM network traffic. The system optimizes the use of display space by allocating information around the inside surface of a ring in 3D space. In the case of multiple protocol layers, it stacks the layers one on top of the other in the vertical or horizontal direction. Through the use of transparency, objects become partially transparent so background objects are visible through foreground objects. With this 3D system, no matter how may protocol layers there are and how complex the underlying structure is, there is still enough space to adequately allocate information in a single display. An important contribution of our 3D visualization system is its ability to provide users with an intuitive presentation of a complex information space, thereby facilitating error detection and performance analysis of ATM networks.

Acknowledgements

I especially thank my supervisor, Dr. Mark Green, who not only made this project possible, but through his constant encouragement kept me on the right track. My appreciation is extended to the other examiners of my thesis, Dr. Janelle Harms and Dr. Desmond Rochfort, for their suggestions and corrections which have been incorporated into the final version. Thanks also go to Dr. Peter van Beek who chaired my thesis defense.

I must also thank all members of the Computer Graphics research group for their support, patience, and encouragement. Special thanks go to both Dr. John Buchanan who always gives me valuable suggestions and comments of my research, and Lloyd White who help me to prepare all demos.

And finally, I would like to thank my boyfriend, Wing Chan, for his support, love, and understanding. Above all, I give thanks to God for giving me everything such as the opportunity, ability, and stamina to prepare.

Contents

List of Tables

List of Figures

List of Plates

List of Symbols

Chapter 1

Introduction

1.1 Why Information Visualization

As we near the end of the century, we find that computer networks have greatly improved the quality of our lives and our working environments. The networking of computer allows the sharing of distant resources and provides an efficient means to transport huge volumes of data among remote sites. It also furnishes interactive communications among network users located geographically apart and allows centralized masters for remote systems to manage day-to-day transaction at banks and perform inventory management at department stores. More importantly, it produces a very flexible working environment. Connecting terminals or portable computers with networks through telephones, for example, allows us to conduct our personal and professional business anywhere.

Given the impact of computer networks on our lives, studying network traffic data is an important first step in understanding, analyzing, and evaluating networks. However, the great variety of data elements (e.g. time, id, and length), the wide range of data types (e.g. text, video, and audio data), and the enormous amount of data flowing over networks make data analysis and performance evaluation difficult. There is an urgent need for effective techniques to help us understand network data.

Information visualization is a promising approach for facilitating human comprehension of complex structures and large volumes of data. The main difficulty in understanding network data is that it always involves large amounts of quantitative information, and the human visual processing capacity is limited. If our eyes and brains allowed us to read and memorize a large volume of data at once, we would

have no difficulty in understanding the data. However, most humans can read at best 1200 words per minute [18], and the written words or text have to be read before they can be understood. This means that textual representation is not an efficient way to understand large amounts of quantitative information. Information visualization takes advantage of the human perceptual ability by transforming data which are not inherently spatial into a visual form. It utilizes a visual representation of data to stimulate human recognition of patterns and structure in information, allowing us to efficiently understand the information through direct observation.

1.2 Problem Definition

The motivation for this thesis is the problem of visualizing ATM network traffic data collected by the Hewlett-Packard (HP) 75000 protocol analyzer. ATM is a set of high speed network protocols that are designed to support a multiplicity of services, such as image, audio, video, and other computer data. Each of these services demands different network characteristics. In order to determine whether the requirements for individual services are being met, the HP 75000 protocol analyzer is used to analyze the performance of devices connected to an ATM network. In this analysis process, up to 8 Mbytes of data, representing over 130,000 ATM cells, are collected by the analyzer and stored in the capture buffer for playback and later analysis. Having efficient visualization techniques for displaying this large volume of data is essential in analyzing network performance.

Text-based visualization is not a good approach for visualizing ATM network data. The technique originally used in the HP 75000 protocol analyzer involved using several lines of text to represent each cell and then displaying consecutive lines of text on a 9 inch text window with some mechanism for scrolling. 130,000 ATM cells require hundreds of thousands lines of text. Due to the size of the window, only a small portion of the structure is shown. Users must navigate through the text to construct an overview of the information structure. In order to detect an unusual pattern or an interesting spot, users have to jump from one page of text to another. For error detection, users need to go through the text line by line. Obvioully, text-based visualization is not useful in visualizing ATM network data.

In [18], Luo proposed a number of two-dimensional (2D) visualization techniques for visualizing ATM network data. One of the problems associated with using text-based visualization techniques to display ATM network data is that users do not observe an overall view of the information space; instead, they must mentally construct the overall view by moving back and forth the information space. Besides, users do not gain any insight from the textual representation. Luo solved these problems by providing users with time-line visualization and cell visualization.

- Time Line Visualization
  - Makes use of hierarchies to display data at several levels of detail. An overview of the data is presented at the top level of the hierarchy, and a detailed view of data is given at the bottom level.
  - Presents the temporal distribution of the data.
- Cell Visualization
  - Makes use of graphical properties to encode important fields of data units.
  - Not only provides more details of each data unit but also emphasizes important characteristics embedded in the data.

Since the amount of ATM network data to be displayed is huge, no matter how compact the representation is, it is impossible to show all the data at the same time on one screen. Luo solved this problem by providing a scrolling mechanism. By scrolling the time-line visualization, users can visualize the complete data space. Applying the scrolling mechanism to cell visualization, users can obtain more details of each data unit.

The combination of both visualization techniques has many advantages over traditional text-based visualizations, including an overall view of the data collected and a visual representation that provides a perspective view of the data. The use of Luo's visualization techniques can facilitate error detection and performance analysis in ATM networks. However, the visualization techniques proposed by Luo are still not good enough to visualize ATM protocol data.

Using Luo's visualization techniques to display multiple protocol levels in ATM networks results in a space allocation problem. All visualization techniques presented in Luo's thesis are based on a two-dimensional (2D) visualization space. Since there is not much room available in the 2D space, the amount of information coherently brought together on the display is limited. The limitation of screen space may not be so serious when only one level of the protocol stack is examined but becomes obvious when multiple levels of the protocol stack are involved in visualization. The use of the hierarchical concept in Luo's visualizations can provide users with multiple levels of detail. However, each level of the protocol stack demands its own timeline hierarchy. For multiple levels of the protocol stack, multiple timeline hierarchies are required. This quickly uses up the available screen space. Due to the limitation of screen space, Luo's visualization techniques may not be appropriate for displaying multiple levels of the protocol stack at multiple levels of detail.

1.3 Thesis Objective

This thesis is based on a visualization project for the HP 75000 Broadband Series Test System. For the last few years, a number of 2D visualization prototypes have been developed. Now, we are extending our visualization techniques to 3D space to solve the space limitation problem in 2D visualizations.

Moving from 2D to 3D visualizations is obviously a good solution to the screen space problem because in 3D space, the extra dimension may provide extra room for placing information and allows two or more relationships among data to be represented at the same time. However, navigating around 3D space is a difficult problem that must be considered in these visualizations. In 3D space, the position and orientation of users affect the aspects of the visualization that are visible. The navigation problem in 3D space requires careful attention in order to develop effective visualization techniques.

The aim of this thesis is to develop a 3D visualization technique that not only solves the space limitation problem encountered in 2D visualizations but also provides users with an efficient way to navigate around the 3D space without becoming any confusion.

The goals of our technique are as follows.

- Make use of graphical properties to highlight characteristics of the data.
  - Users can easily detect interesting spots and recognize unusual patterns.
- Provide continuously variable levels of detail for the data.
  - Users can obtain an overview of the data set and a detailed view of an individual data unit.
- Make use of several 3D interaction techniques to simplify the navigation problem and provide a direct manipulation mechanism.
  - Users can effectively navigate around the information space and can earn greater insights into the data by manipulating the display.
- Take advantage of the extra degree of freedom to place more information in the visualization.
  - Users can receive more important information from a single screen.

Our goal is to investigate 3D visualization techniques that can provide an efficient and intuitive presentation of a large information space, thereby facilitating error detection and performance analysis of ATM networks.

1.4 Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2. we first review some previous works in the area of information visualization. In Chapter 3, we give an overview of ATM networks. In Chapter 4. we identify all problems associated with the visualization of ATM protocol data. In Chapter 5, we present our approaches to overcoming the problems identified in Chapter 4. Chapter explains the implementation details of our visualization techniques and describes our phototype system. In Chapter 7, we compare our system with the system developed by Luo [18]. Finally, in Chapters 8 and 9, we conclude and provide directions for future work.

Chapter 2

Related Work

A particular problem associated with visualizing large volumes and complex structures of data is inadequate display space. Even if the size of the data we work with becomes large, the window on which the data is displayed remains small, and only a small portion of the data can be visible at once. Since the position of local information in a global context can determine the importance of the information, it is crucial to visualize where the local information is in the global picture. Therefore, the amount of information that can be visible at once affects how easily the information can be understood. Information visualization not only utilizes visual representations of data to facilitate human comprehension but also includes some techniques to solve the space limitation problem. The purpose of this chapter is to review a variety of information visualization techniques that have been developed for maximizing the efficient use of small screen space.

The techniques described here can be roughly classified into three categories. They are zooming and filtering, focus and context, and widgets for information visualization. The Starfield Display [14], and Pad++ [5] can be classified as zooming and filtering techniques. The Perspective Wall [19], Cone Tree [27], Information Cube [26], Hyperbolic space [21, 16], Rubber Sheet [28], and Fisheye View [10] can all be classified as focus and context techniques. Finally, the Alphaslider[4] and Magic Lens^{TM} [6, 30] are described as interactive techniques for information visualization. These classifications are only rough. Some techniques may fall into more than one category. For example, the Magic Lens^{TM} may also be placed into the focus and context category.

The following sections describe each of these categories and techniques in detail.

2.1 Zooming and Filtering

The first approach to the problem of space allocation is to reduce the amount of content in the display by zooming and filtering. Sometimes the amount of information available makes it impractical to display all of it. For example, displaying all information may induce a serious occlusion problem and degrading the response time of the system. The data have so many dimensions that it is impractical to display all of them at once in a 2- or 3-dimensional display. Users know they are interested in only a particular subset of the data. One solution is to allow users to flip between different views at different levels. In this way, one can start out at a global but very low detail view and then zoom in selectively on different local regions. In order to know where the current view fits, he or she can step back by zooming out to view the global structure of the file. Zooming and filtering the information in some way can reduce the amount of content in the display. Techniques that fall in this category include the Starfield Display[14] and Pad++.

Starfield Display

The Starfield Display[14] selects two ordinal attributes from a multi-attribute database and encodes the selected attributes as coloured glyphs on a two-dimensional display. It provides interactive smooth zooming to facilitate rapid information browsing and searching. Traditional zoom techniques are used to move away or towards a focal point in all directions, whereas the zooming technique used in starfield displays continuously changes the range of attributes along either axis individually. The essential advantage of this zooming technique is its ability to incrementally and smoothly zoom in on an area of interest by changing the range and scale of each axis, thereby allowing users to track the motion of the glyphs without suffering disorientation as a result of sudden, large changes in view. In addition to the ability to alter the range of each axis, it can filter out unwanted items and reduce clutter. Coupling these strengths with the starfield display can facilitate the tasks of information browsing and searching.

Pad++

Pad++ [5] applies several interactive interface mechanisms to the problem of information presentation. A particular problem associated with information presentation is how to make information searching and browsing more efficient and intuitive. This problem is addressed in Pad++ by three mechanisms. Most importantly, Pad++ allows users to zoom into any area of the content at will. The zooming technique used in Pad++ provides users not only with a scaled down or up version of an object but also with a different representation of the object. For example, when zooming into a map, Pad++ shows continents, and then countries, towns, and so on. When zooming out, Pad++ simply displays five continents. Lenses are used to modify the appearance and behaviour of objects so that users can receive more intuitive presentations of information. Finally, Pad++ allows users to compare two distant regions, or look at an overview of a document together with its closer view. Portals, special to Pad++, are like holes through which users are able to look at other regions of the Pad++ surface or even at other Pad++ surfaces. When an object is seen through a portal, the portal can pass interaction events to the object. Combining portals with lenses allows users to interactively change the nature of the data underneath them.

2.2 Focus and Context

The second approach to the problem of maximizing the efficient use of display space is focus and context. This technique allows users to view a small central focus while maintaining the visibility of a larger context. Zooming and filtering are used to make the display more desirable by reducing the amount of information on the display. The main drawback of this approach is that local and global information are not available at once. Focus and context is a technique which allows users to examine a particular area in detailand at the same time provides them with the overall view of the information structure.

Fisheye View

Furnas presented a distortion technique called the Fisheye View [10] which provides users with both the "global context" and "local detail" simultaneously. A fisheye lens

is a very wide angle lens. It shows nearby regions in great detail and surrounding regions in successively less detail. Applying this lens to a graphical display of information enlarges a certain region around the point of interest while at the same time covering a large field of view, so users can visualize an entire information structure at once as well as zooming in on specific items.

The main problem of Furnas's fisheye view is that the transition between the contextual view and detailed view is too sudden. Fisheye views are generated by Degree of Interest functions that are thresholded to determine the contents of the display. However, thresholding causes the visualization to have gaps that might be confusing or difficult to repair. Furthermore, gaps can make it difficult to change the view. The desired destination might be in one of the gaps, or the transition from one view to another might be confusing as familiar portions of the visualization suddenly disappear into gaps.

Rubber Sheet

The rubber sheet, described by Sarkar *et al* [28], is a technique for viewing large and complex layouts within small display areas. With this stretching technique, user can stretch different areas of the sheet with a set of handles. As the user stretches an area, a greater level of detailed is displayed there. In this case, the user not only can select an area of interest but also can control the level of detail of the area displayed. There are several desirable features provided by this technique: it allows exact specification of focus, provides uniform scaling at focus, integrates focus and context, allows precise space allocation, preserves overall shape, and allows multiple foci.

The Perspective Wall

The Perspective Wall [19], described by Mackinlay *et al*, is a 3D visualization technique for visualizing linear information. After performing some case studies, Mackinlay *et al* concluded that information spaces often contain spanning properties such as time. The most common way of visualizing these information spaces is to structure them linearly. These linear structures result in 2D layouts with broad aspect ratios that are difficult to fit in a single view. In order to put all the information on a small computer screen, layouts with broad aspect ratios must be reduced in scale, which

leads to a reduction in level of detail. When the display is enlarged, part of the layout must be sacrificed. There is a tension between detail and context. Mackinlay *et al*'s Perspective Wall is a graphical device which integrates details and context of a linear structure into a single view. The idea is to draw the information onto a long horizontal rectangle, fold the rectangle back at the left and right boundaries of the area of detailed interest, and then generate a 3-D perspective image of the result. Any desired amount of context can be made visible by adjusting the angle of the folds, from zero degrees to an angle somewhat less than ninety degrees. The advantages of this visualizations are that it uses folding to distort an arbitrary 2D layout into a 3D visualization, which provides efficient space utilization for 2D layouts with wide aspect ratios, it provides an intuitive 3D metaphor for distorting 2D layouts, and it allows for smooth transitions among views.

Cone Tree

The Cone Tree presented by Robertson, Mackinlay and Card is a 3D visualization technique for visualizing hierarchical information [27]. In the Cone Tree representation, hierarchical information is depicted as a collection of 3D cone-shaped structures. Each cone corresponds to one subtree in the hierarchy, with one node located at the apex of the cone and a group of subnodes arranged evenly around the rim of the cone. By using this representation, the whole hierarchy, or a large portion of it, can be displayed on a computer screen. Thus, users no longer rely on scrolling to view the overall structure. In order to focus on any interesting subtree in detail, users can utilize animation provided by the system to rotate that particular part to the front, while still keeping the context of the entire tree. The Cone Tree approach not only provides users with an overview of the entire information structure, but also allows users to focus on different portions in detail. Therefore, users can find the relationship between a particular part to the whole tree.

The problem with the Cone Tree approach is that the technique is not entirely satisfactory for visualizing a huge hierarchy due to the visual clutter problem [26, 27]. Once the tree to be displayed is balanced or exceeds 1000 nodes, the resulting image becomes complex and cluttered [26, 27]. Certain interesting nodes in the tree may be occluded by the nodes and edges at the front. Users have to rotate one subtree to the

front and check each node in the subtree to find an interesting node. This occlusion prevents users from gaining a perceptive view of the information space and makes it difficult to detect interesting spots. Due to the increasing visual clutter, the Cone Tree technique is limited to visualizing small hierarchies and is not appropriate for a huge and dense hierarchical structure.

Beyond Cone Trees

Carriere and Kazman enhanced the Cone Tree approach [27] to develop a better visualization system, called fsviz [8]. In order to effectively display a hierarchy, the Cone Tree approach places a lot of restrictions on the size of the hierarchy, for example, fewer than 1000 nodes, and no more than 10 layers, [26, 27, 29]. These restrictions are addressed in fsviz by three mechanisms. Most importantly, fsviz eliminates the visual clutter by applying an intelligent node layout algorithm and by incorporating techniques such as an automatic coalescing of subtrees and fisheye viewing for reducing the complexity of the resultant image. Fsviz also allows users to directly manipulate both the overview of the tree structure and the orientation of the subtree through the use of the hand-coupled rotation. In addition, the automatic animation provided by the system not only rotates a node to the front of the tree but also brings users to the selected set of nodes. Finally, fsviz supports the use of dynamic queries, allowing users to specify which part of the tree is to be pruned or expanded so that users can control the class of information visualized. With Carriere and Kazman's fsviz, users initially obtain an arbitrary but intuitive view of the hierarchy, and through the use of automatic animations, hand-coupled rotations, and dynamic queries, rapidly locate the portion they are interested in.

The Information Cube

Rekimoto and Green's Information Cube is another visualization technique for hierarchical structures [26]. The Information Cube makes use of translucent and nested cubes to denote hierarchical information. The outermost cube corresponds to the data at the top level of the hierarchy, and all cubes placed inside this outermost one represent the data below the top level. Any inner cube, which corresponds to lower level data, must be enclosed by an outer cube, which corresponds to higher

level data. In conjunction with translucency, all cubes become partially transparent to users so that inner cubes are visible through outer cubes. By making the cube semi-transparent, users can not only visualize the whole data set but also distinguish higher and lower level data clearly and easily. With the use of several 3D interaction techniques provided by the system, such as DataGlove and EyePhone, users can navigate and manipulate the data stored in the information cube effectively.

Table Lens

Traditional spreadsheets provide users with mechanisms to present tabular information. These mechanisms are easy to use, but not efficient, especially when the database is huge. The Table Lens [31] supports efficient visualization of huge tabular information. It combines focus+context with a graphical mapping scheme to enable exploratory data analysis. This technique has many advantages over traditional spreadsheets, including visual representations of collections of values that provide users with an insightful sense of the data and fisheye views that allow interaction with large information structures by dynamically distorting the spatial layout of the structure according to the varying interest levels of its parts. However, the technique still has a few limitations: it is restricted to visualization of tabular information and provides only one type of distortion.

Hyperbolic Browser

3D visualization techniques are appealing, especially for visualizing huge and complex hierarchies, because the extra dimension in 3D space provides extra space for visualization, thereby alleviating the space limitation problem. However, this does not imply that 2D visualizations cannot be as good as 3D techniques in solving the problem of space limitation.

In [16], Lamping *et al* presented an efficient 2D visualization technique, called the hyperbolic browser, to display and manipulate large hierarchies. The exponential growth of the tree makes the display become cluttered very quickly because linearly increasing the area in Euclidean space does not provide sufficient space for an exponentially expanding tree. However, in hyperbolic space, the area of the circle increases exponentially with its radius, which means that the space available grows

exponentially with its distance. Thus, hyperbolic planes can provide more room for visualization than Euclidean space. Furthermore, given a line and a point not on the line in hyperbolic space, there is more than one line passing through the point and parallel to the given line. This special behavior greatly reduces the overplotting problem in visualization. Lamping *et al* took advantage of these properties of hyperbolic space to solve the visual clutter problem. The basic idea underlying their technique is to lay out a tree on a hyperbolic plane with the root at the center of the plane and then project the plane onto a circular display screen. The aim of arranging a tree in a hyperbolic plane is to obtain a lucid layout of the tree before projection. The projection allocates more space to the portions near the center of the plane and reduces the space allocated to the nodes away from the center. This means that regions near the center become enlarged, while regions away from the center diminish. In order to bring any component of the tree into focus, the user simply selects the component, and the selected part is brought into focus at the center. The hyperbolic browser provides users with the ability to visualize an entire hierarchy and focus on specific regions concurrently. Therefore, users can search for patterns in the big picture and investigate any interesting details without losing context.

3D Node and Link Display

Traditional node and link 2D displays are good for small and scattered networks but become inadequate for any reasonably sized ones. These 2D displays consist of a collection of nodes representing the objects in the network connected by lines representing the traffic among objects. Such displays work well for small and dispersed networks but lose their efficacy once the network size becomes large and dense. The main reason is that the long lines connecting remote pairs of nodes cause overplotting in the display. This overplotting is a serious problem because nodes may be completely blocked by the crossing lines and thus appear unclear. The overplotting makes it difficult to clearly perceive the network data, let alone to detect any pattern in the display.

In order to eliminate the visual occlusion problem with 2D displays, Cox and Eick proposed a 3D visualization technique for displaying network data [9]. In this technique, nodes are positioned on a sphere, and arcs, instead of straight lines, are

drawn among nodes to represent the traffic in the network. The line crossing clutter problem can be reduced because the extra dimension in 3D space provides arcs with various possible heights. The height of the arc depends on the significance of the arc. The more important the arc, the larger the value of the height assigned to it. In this way, all significant arcs are apparent to users and are never blocked by less significant ones. In addition, some user interface controls, such as thresholding and translucency, used in the technique allow users to interactively select the opacity of the sphere, thereby adjusting the complexity of the display.

3D Hyperbolic Space

There is no doubt that Lamping *et al*'s hyperbolic browser is an efficient technique for visualizing large hierarchies. However, their approach deals only with acyclic graphs, i.e. graphs with no cycles.

In [21], Munzner and Burchard applied 3D hyperbolic geometry techniques to the problem of visualizing cyclic graphs, i.e. a tree structure with links back up to the tree. They addressed the fact that many tree hierarchies are not simply acyclic graphs, but are graphs containing many cycles. Consider, for example, Unix file systems with symbolic or hard links. In order to accurately represent these tree structures, Munzner and Burchard took account of cycles in their visualization. They initially laid out a tree in a 3D hyperbolic space with no "backlink" and then filled in the "backlink" edges.

2.3 Interaction Techniques

Along with visual display techniques, information visualization has a set of tools for interacting with large amount of information. These techniques allow users to select a focus, filter out unwanted information, zoom in on certain ranges of information, and zoom out to visualize the global picture. The techniques falling into this category include the Alphaslider [4] and Magic Lens [6, 30].

Alphaslider

Ahlberg and Shneiderman's alphaslider [4] allows users to locate a particular item from a long list in an easy and rapid manner. Using the traditional scrolling mechanism, a scrollbar is used to control which portion of the list is visible. Users first page through the list to look for a particular item and then select the item by clicking on it. The ease of locating a particular element depends on the size of the display screen. As the size of the window increases, the amount of information that can be visible at once increases, thereby decreasing the difficulty of locating a particular item. The Alphaslider improves this selection task. It reduces the amount of screen space required and increases the speed of item searching. It consists of a text output, a slide area, a slider thumb, and an index to the elements that the slider operates over. The slide area is a small rectangular bar. The index below the slide area is proportionally spaced to the number of items that start with each character. Any starting character of an item which does not appear frequently will not be shown in the index. Instead of paging through the list to look for a particular item, the alphaslider allows users to move directly on a certain part of the slider range by either clicking on it or dragging the slider thumb. The output of alphaslider is shown as a single line text item, which is updated as users move the slider thumb.

Magic Lens

In [6, 30], Stone *et al* presented a new graphical filter, called the Magic Lens^{TM} filter, for exploring various types of information and providing users with various distortions of information. A magic lens filter is a movable, arbitrarily shaped region, plus a filter which modifies the view of the region underneath the lens. Different types of filters have different effects on the appearance of the region. Overlapping filters can compose their effects on the overlapped region. The advantages of using the Magic Lens^{TM} filter arethat it not only shows the modified view of the selected region but also keeps the region in the context of the whole view. Besides, it provides users with different perspectives of the viewing regions, and can be used with various types of information.

Chapter 3

A Brief Overview of ATM

In this chapter, we provide the reader with some background on ATM, including ATM technology, the ATM protocol reference model, and ATM networks so that the reader gains sufficient knowledge to understand the remainder of this thesis. The intention of this chapter is not to give a complete overview of ATM but to introduce some key concepts that influence the selection of visualization techniques. A more detailed description of ATM can be found in [1, 2].

3.1 ATM Technology

While the needs for multimedia applications have been increasing over the last few years, the next big jump forward in demand is anticipated to be an efficient networking technology for the transport of various types of services over the same network.

Some telecommunication networks are capable of dealing with many useful applications; however, they were not originally designed to handle emerging multimedia applications. In order to carry various types of services on a single network, the network must be able to integrate and transport all types of services with various traffic characteristics and performance requirements. The telecommunication networks developed in the past were mainly designed for the support of one particular service and cannot be efficiently used for purposes other than the ones they were designed for. They successfully support many useful applications but cannot inadequately cope with the needs of new multimedia applications.

In an attempt to have a networking technology support not only one particular service but any kind of service in an integrated manner, Broadband Integrated Ser-

vices Digital Network (B-ISDN) was introduced. B-ISDN is a service-independent network in which all kinds of services with different traffic characteristics and performance requirements can share the same set of network resources. The emergence of B-ISDN meets the demand for efficient networking technology to support new services in an integrated manner.

Asynchronous Transfer Mode (ATM) was chosen as the switching and multiplexing techniques for B-ISDN. ATM combines the benefits of packet switching with those of multiplexing to provide greater flexibility and efficiency in B-ISDN. In the following two subsections, we discuss how packet switching and multiplexing techniques facilitate the transport of various services over ATM networks.

3.1.1 Fast Packet Switching

In ATM networks, service data are first divided into smaller, fixed-size, and standardized units, or packets, before transmission. These packets have sufficient control information attached to them to identify the packets belonging to the same service, so that they can be routed through the network independently.

The essential advantages of using smaller, fixed-size, and standardized packets in switching are their abilities :

(1) to support various kinds of services in an integrated manner

> Due to the fact that all packets are the same size with standardized structures, no matter what type of service data a packet contains, the network and the switches can blindly forward all packets to their destinations without having to understand each separate type of service.

(2) to have the shortest possible delay and highest throughput

> For a network with smaller packets, the time required to switch a packet is shortened, thereby reducing the waiting time or the delay for the next packet. Therefore, if a high priority packet comes next to a low priority packet, the former will need to wait only for a short period, instead of the longer time required to process a larger data unit. In addition, many packets from the same service can be in transmission concurrently, thereby speeding up an entire data

transmission process.

(3) to simplify the delivery mechanism

Packets with variable length require complex buffer management inside the network. In order to have a satisfactory performance, sophisticated protocols are necessary to perform error and flow control on every link of the connection. The more sophisticated the protocol is, the higher the processing requirements and the longer the delays inside the network, thereby complicating the entire delivery mechanism. With fixed-size packets, different traffic types can be delivered on the same network in a predictable manner and by the same mechanism.

(4) to lower the programming cost

Every packet has a standardized structure, 5 bytes for the header and 48 bytes for the payload. The information in the header is used to guarantee the proper routing of each packet in the network, while the payload carries only the user service data. Therefore, only information in the header needs to be updated in each station of a connection for an entire switching process. Due to the limited functionality in the header, which we will discuss in section 3.3.3.1, processing of the header is very simple and can be done at very high speed. Thus, the cost of the header construction can be lower.

3.1.2 Asynchronous Time Division Multiplexing

A multiplexer enables a number of independently generated packets to be interleaved and transmitted on a single traffic stream. This step may be accomplished in a synchronous or an asynchronous manner. The one used in ATM is Asynchronous Time Division Multiplexing (ATDM).

ATDM allows more efficient use of communication channels than does Synchronous Time Division Multiplexing (STDM). In STDM, each station is preassigned a fixed time slot on a communication channel and can have the full use of the channel during that preassigned time slot. The main problem with STDM is that it does not make good use of the bandwidth. No matter whether a station has anything to send, a time slot has been preassigned to the station and cannot be used for other stations.

Since not all stations always have something to send, a station with nothing to send will waste its preassigned time slot. ATDM is intended to solve this problem. In ATDM, instead of preassigning a time slot to each station, a time slot is assigned only if a station requests it. Therefore, each station can transmit service data as long as it needs to and empty time slots are available. With ATDM, each time slot on a communication channel can be used by any station. Which station can get access to the channel depends on both the order of the request and the priority of the station. It is obvious that ATDM can more efficiently utilize a communication channel.

3.2 ATM Protocol Reference Model

ATM is a layered architecture in which different types of information, such as voice, video, image, and text can be integrated and transmitted over a single network. It consists of three functional layers: a physical layer, an ATM layer, and an ATM adaptation layer. These three layers form a stack which is called the ATM protocol stack, as shown in Figure 3.1. In the following three subsections, we review the functions of these three layers.

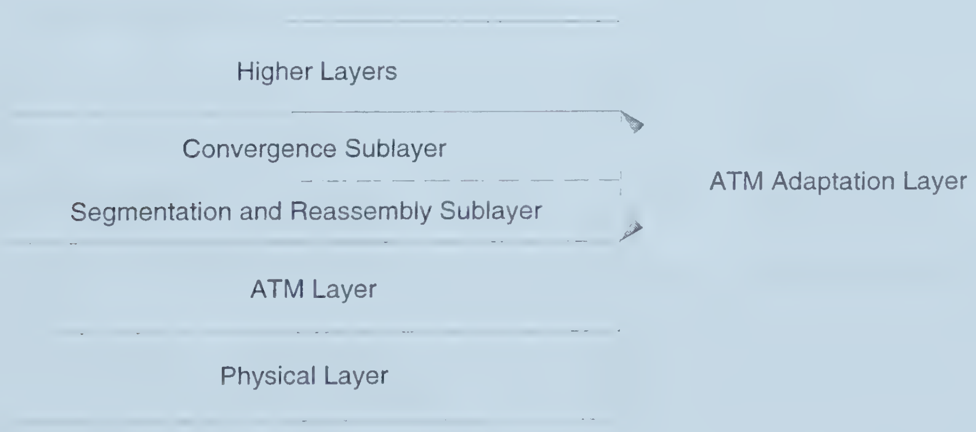


Figure 3.1: ATM Protocol Reference Model

3.2.1 Physical Layer

The physical layer is at the lowest level of the protocol stack. It is used to :

- encode or decode data into appropriate electrical formats so that the data can be transported between two ATM entities.
- present some cell delineation indications to the ATM layers so that the cell boundaries can be recovered in the receive direction.
- generate and verify the header error control field. If any error cell is detected, the cell will be either corrected or discarded. The physical layer guarantees that only valid cells are entered into the ATM layers.
- insert or remove idle cells so that the rate of the ATM cells to the payload capacity of the transmission system can be adapted.

3.2.2 ATM Layer

Immediately above the physical layer is the ATM layer which is responsible for :

- extracting ATM cell headers so that only the cell payload enters the ATM adaptation layer, which is directly above the ATM layer.
- adding appropriate cell headers (minus header error control field) to the cells from the ATM adaptation layer.
- updating VPI and VCI values. All cells must have their VPI and VCI fields updated before they are transmitted to the next ATM switch or end system. A brief description of VPI and VCI will be provided in section 3.3.3.1.
- multiplexing cells from separate VPs and VCs into a single cell stream.
- demultiplexing a single incoming cell stream into several individual cell streams according to the VPI and VCI.

3.2.3 ATM Adaptation Layer

The layer above the ATM layer but below the higher layers is the ATM adaptation layer (AAL). As shown in Figure 3.1, the AAL is divided into two sub-layers :

Segmentation and Reassembly (SAR):

This layer is responsible for :

- segmenting service data from the higher layer into several 48 byte units which serve as the payloads of the ATM cells.
- reassembling multiple ATM cell payloads into a single service data unit which will be transmitted to the next higher layer.

Convergence sublayer (CS):

This layer is responsible for :

- offering a group of functions to support a specific service.
- inserting additional information required for adaptation to the higher layer data unit.
- providing the network user with an interface.
- This layer is further divided into two sublayers: the Common Part Convergence Sublayer (CPCS) and the Service Specific Convergence Sublayer (SSCS) to assist data transportation.

3.3 ATM Networks

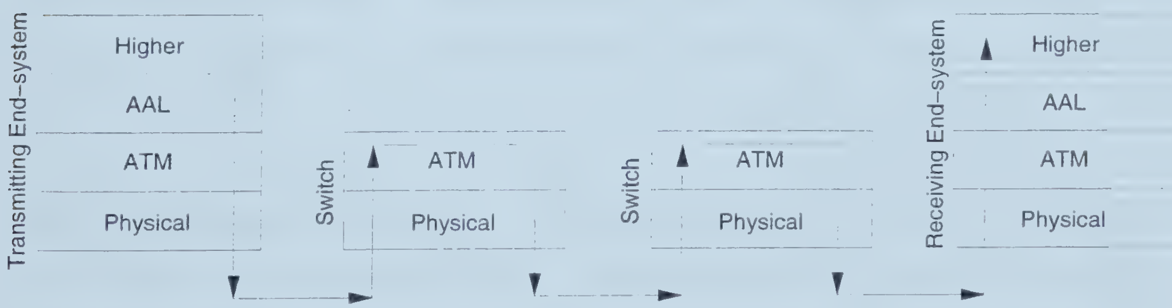


Figure 3.2: An ATM Network

An ATM network consists of a collection of ATM switches and ATM end-systems that are interconnected by communication channels. Figure 3.2 shows a very simple ATM network model in which two end-systems and two switches are involved. After a connection has been set up between two end-systems, service data can be sent from one end-system to the other. In this section, we will concentrate on how two end-systems can exchange service data through an ATM network.

3.3.1 ATM Network Service

Using ATM, service data are segmented into smaller, fixed-size, and standardized packets, or cells. These cells are then transported to and reassembled at the receiving end-system.

At the transmitting end-system, service data traverse through the protocol stack, from the higher layers to the physical layer. At each layer of the protocol stack, a protocol data unit (PDU) from the upper layer is segmented into several smaller units. Each of these units has sufficient control information attached to it. The unit together with the attached information, becomes one PDU to the next lower layer. As each PDU comes down to the ATM layer, a collection of ATM cells, which will be discussed in section 3.3.3.1, are generated and are passed onto a communication channel.

ATM switches are based on the routing information contained in the ATM cell header and the information stored inside the switches to switch cells from incoming channels to outgoing channels. They are responsibe for updating the cell header information.

As an ATM cell arrives at the receiving end-system, it comes up the protocol stack in the reverse order it went through from the transmitting end-system. At each layer of the stack, the control information added at the equivalent layer of the transmitting end-system is stripped off, and PDUs that belong to the same next higher level PDU are reassembled. Therefore, as PDUs move up the protocol stack, several lower level PDUs are reassembled to produce a single higher level PDU; finally, the original service data are constructed at the higher layers. In this way, the receiving end-system can receive the data that are sent by the transmitting end-system.

3.3.2 ATM Cell Preparation

In the entire data exchange process, ATM cells act as the basic unit. Figures 3.3 and 3.4 show how AAL 3/4 and AAL 5 prepares ATM cells for transmission.

As shown in Figure 3.3, a PDU at the convergence sublayer (AAL 3/4 CPCS-PDU) is first created by inserting a header and a trailer to the data frame. Then the segmentation and reassembly sublayer segments the AAL 3/4 CPCS-PDU into

smaller PDUs and attaches a header to each PDU segment. Each PDU segment and its header become a PDU in the SAR sublayer, which is called an AAL 3/4 SAR-PDU. This SAR-PDU will be the payload of an ATM cell. At the ATM layer, a header is attached to each SAR-PDU to form an ATM cell.

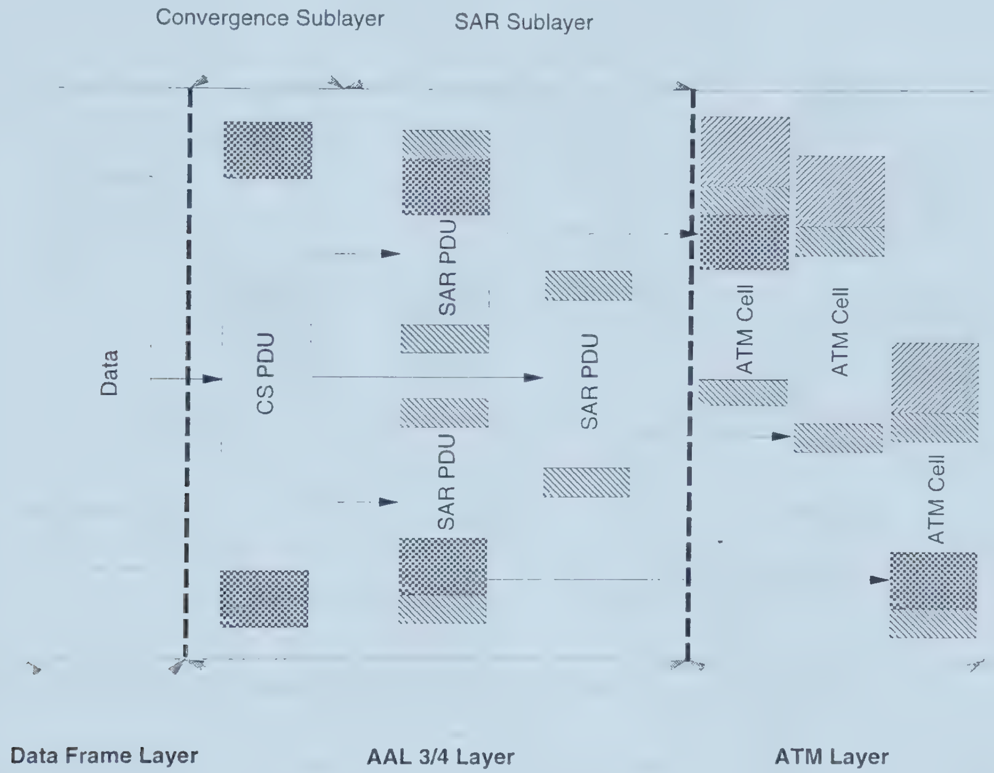


Figure 3.3: AAL 3/4 Cell Preparation

Figure 3.4 shows how AAL 5 prepares ATM cells for transmission. The convergence sublayer creates a PDU, called an AAL 5 CPCS-PDU, by attending a variable-length trailer to a data frame. The pad in the trailer guarantees that the CPCS-PDU is long enough to fall on the 48 byte boundaries of the ATM cells. The segmentation and reassembly sublayer fragments each CPCS-PDU into several 48 byte units. Each such unit is called an AAL 5 SAR-PDU. The entire SAR-PDU will become the payload of an ATM cell. The header of the ATM cell will be generated at the ATM layer and attached to a SAR-PDU to form an ATM cell.

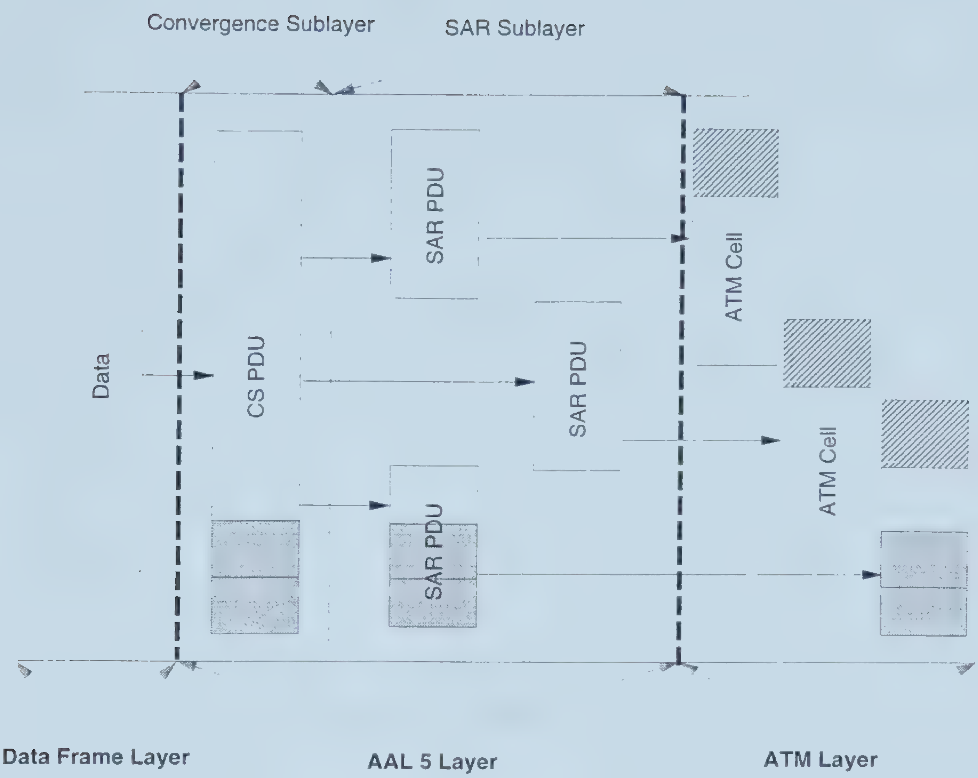


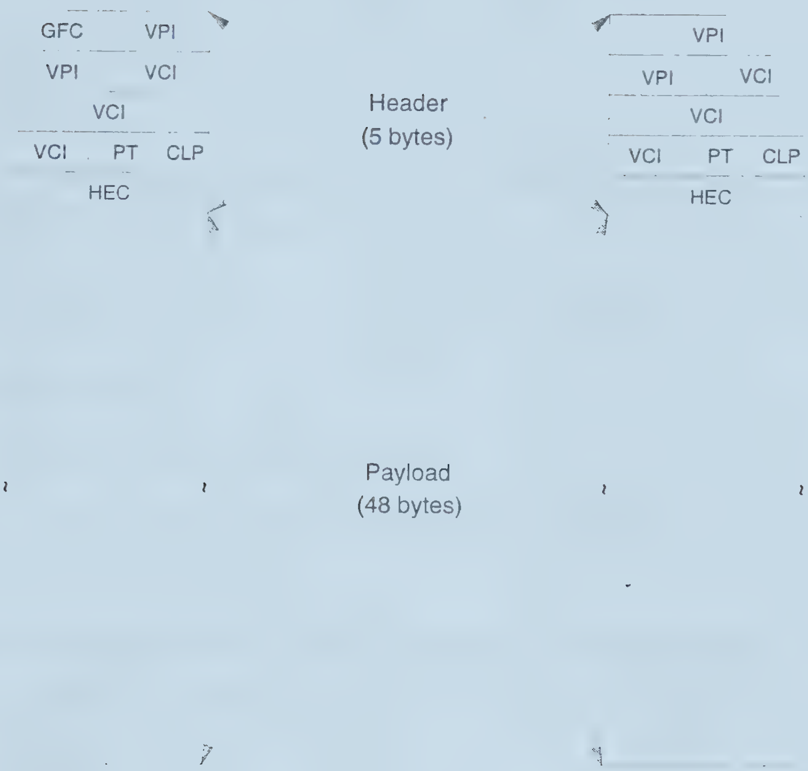
Figure 3.4: AAL 5 Cell Preparation

3.3.3 ATM Protocol Data Structure

The previous subsection focused on ATM cell preparation. We notice that many different PDUs are produced at different levels of the protocol stack. In this subsection, we take a close look at different PDUs and an ATM cell, including examining their structures and the fields in each PDU so that the reader can have an idea of what information may be involved in our visualization.

3.3.3.1 ATM Cell Structure

ATM consists of a sequence of fixed length packets, called ATM cells. Each cell is 53 bytes long, with 5 bytes for the header and 48 bytes for the payload, as shown in Figure 3.5. The header is used to distinguish between ATM cells which belong to different virtual channels, and thus is very useful in network routing. The payload is used to carry ATM service data, and it is relatively less important than the header in performance evaluation. The various parts of the cell header are briefly described as follows.



GFC: Generic Flow Control
VCI: Virtual Channel Identifier
VPI: Virtual Path Identifier
CLP: Cell Loss Priority
HEC: Header Error Control
PT: Payload Type

Figure 3.5: ATM Cell Structure for UNI* (left) and NNI** (right)

* UNI stands for user-to-network interface that defines communications between ATM end-systems and ATM switches.

** NNI stands for network-to-network interface that defines communications between ATM switches.

Generic Flow Control :

- 4 bits in UNI and 0 bit in NNI.
- It is used to regulate cell flowing during periods of network congestion.

Virtual Channel Identifier :

- 16 bits in UNI and NNI.
- It is used, together with the VPI, to determine the next destination of a cell in network routing.
- Each virtual channel can carry only one connection at one time.

Virtual Path Identifier :

- 8 bits in UNI and 12 bits in NNI.
- It is used, together with the VCI, to identify the next ATM switch of a cell in network routing.
- Each virtual path may have a collection of virtual channels; and therefore, cells which are in different virtual channels can have the same virtual path identifier.

Cell Loss Priority :

- 1 bit in UNI and NNI.
- It is used to state the priority of the cell. If there is extreme network congestion or if all buffers are occupied, some cells must be dropped. The cells with low priority must be dropped before any high priority cells are.

Header Error Control :

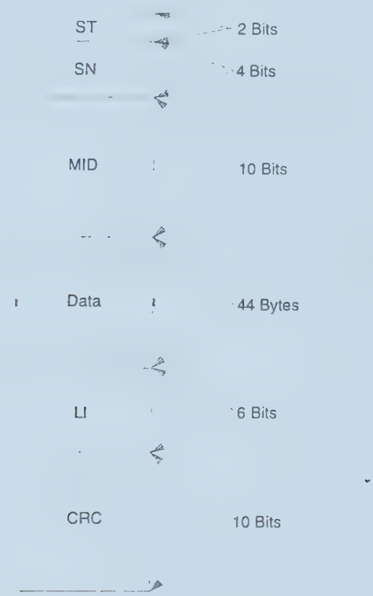
- 8 bits in UNI and NNI.
- Although it exists in the ATM cell header, it is not used by the ATM layer. Instead, it is used by the physical layer for single bit error corrections or multiple bits error detection.

Payload Type :

- 3 bits in UNI and NNI.

- It is used to distinguish between user information and connection management information cells.
- It can also be used to indicate congestion in the network.

3.3.3.2 AAL 3/4 SAR-PDU Structure



ST: Segment Type
SN: Sequence Number
MID: Multiplexing Identification
LI: Length Indicator
CRC: Cyclic Redundancy Check

Figure 3.6: AAL 3/4 SAR-PDU structure

Segment Type :

- It is used to indicate the position (Beginning, continuation, or end of a message) of the unit.

Sequence Number :

- It is used to indicate the order in which SAR-PDUs should be reassembled into a single CPCS-PDU.
- It can be used to check whether misinsections or losses of units occur as PDUs move through the network.

Multiplexing Identification :

- It is used to distinguish between PDUs which come from different traffic sources but interleave on the same virtual channel.
- It is very helpful in reassembling correct PDUs into a single data unit at the destination.

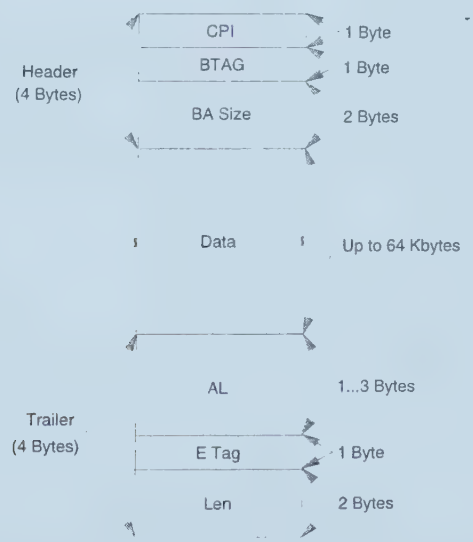
Length Indicator :

- It is used to indicate the size of the payload.

Cyclic Redundancy Check :

- It is used to protect the SAR-PDU by recovering any errors detected.

3.3.3.3 AAL 3/4 CPCS-PDU Structure



CPI:	Common Part Indicator
B Tag:	Beginning Tag
BA Size:	Buffer Allocation Size
Al:	Alignment
E Tag:	Ending Tag
Len:	Length

Figure 3.7: AAL 3/4 CPCS-PDU structure

Common Part Indicator :

- It is used to interpretate other fields.

Beginning Tag and Ending Tag :

- Since the AAL 3/4 CPCS-PDUs are not the same length, the Beginning Tag and Ending Tag are used to fix the boundaries.

Buffer Allocation Size :

- It is used to decide how large a memory buffer must be allocated by the receiver.

Alignment :

- It is used to create a 32 bits trailer, especially when the trailer is less than 32 bits.

Length :

- It is used to determine the size of the packet.

3.3.3.4 AAL 5 SAR-PDU Structure

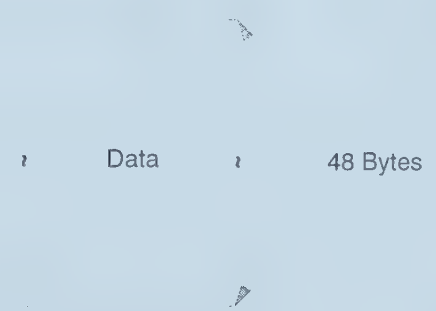


Figure 3.8: AAL 5 SAR-PDU structure

The structure of this PDU is very simple. It consists of only 48 bytes payload data.

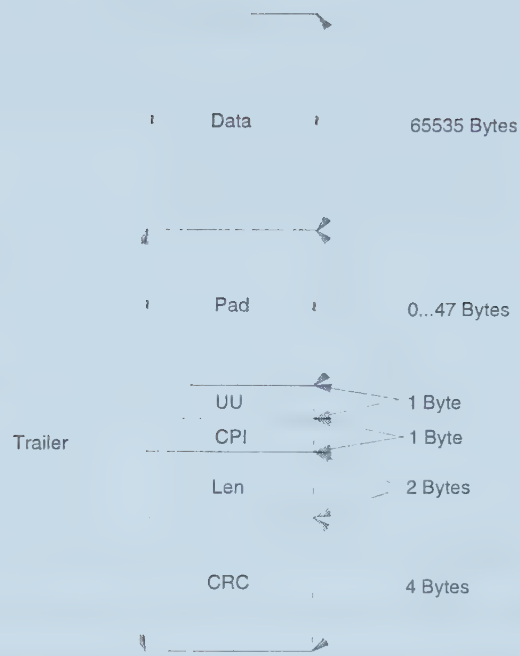
3.3.3.5 AAL 5 CPCS-PDU Structure

User to User Indication :

- It is used to identify the purpose specified by the upper protocols.

Common Part Indicator :

- It is unused and is filled only with zeros.



UU:	User to User Indication
CPI:	Common Part Indicator
Len:	Length
CRC:	Cyclic Redundancy Check

Figure 3.9: AAL 5 CPCS-PDU structure

Length :

- It indicates the length of the packet.

Cyclic Redundancy Check :

- It checks the entire packet.

Chapter 4

Problems

Information visualization systems present a graphical representation of the data and provide users with some techniques for interacting with the information in a large and complicated information space. These systems make important patterns and relevant anomalies easily recognizable and enable users to extract useful information hidden in the information space. Designing such powerful systems for visualizing ATM protocol data, however, is not a trivial task. The fundamental problems involve inventing an adequate representation of the data, and helpful interaction techniques. In the rest of this chapter, we discuss these two problems in detail so the reader can have a good understanding of the motivations behind our visualization approaches.

4.1 Visual Representations

One of the problems associated with ATM protocol data visualization is how should the data be represented in order to facilitate human comprehension. There are a number of layers in ATM, and the user may be interested in multiple layers at the same time. Therefore, several different types of data will be involved in visualizations. The challenge is to design adequate representations for different protocol data so that users can easily recognize different data, understand the relationship between multiple layers of the protocol stack, and how it affects the performance of ATM networks.

The problem of representing ATM protocol data can be further subdivided into three smaller problems.

1. How to represent various types of data.

2. How to allocate space for a collection of ATM cells on a small display window.

3. How to make multiple layers of the protocol stack data visible at the same time.

4.1.1 Representations of Various Types of Data

One efficient approach for concisely conveying information about various data items and their relationships involves graphical representations of the data. Since users are always interested in multiple layers of the protocol stack, the display will have more than one type of data. It seems natural to use graphical representation techniques to gain insight into the data so that their relationship can be easily understood. Graphical representations of the data mean using graphical properties, such as color, shape, size, and position, to represent data attributes and rendering the result on an output window. Representing data graphically highlights important features of the data and makes complex relationships visible. Since humans are excellent at extracting information, recalling related images, and identifying anomalies from graphical displays, graphical representations of the data are useful for helping users better understand the data.

However, finding useful and intuitive graphical representations for depicting ATM protocol data is surprisingly difficult. There are various problems involved.

1. What kinds of information about ATM protocol data should be presented in visualizations?

   Since the data encapsulate a relatively large number of attributes, displaying the data along with all their attributes will affect:

   - the intuitiveness of the display,

     Due to the fact that the screen space is too limited to display a large amount of information, the number of data attributes displayed should not be large, or else the display becomes cluttered, and the intuitiveness of the display will be diminished.

   - the responsiveness of the system, and

     The graphical hardware is limited in its capability to compute and render images. The more images displayed, the slower the system will be.

- the opportunity of gaining insight.

  Humans are limited in their ability to divide their attention among multiple attributes. As the number of attributes to be displayed increases, the time available for focusing on each attribute decreases, thereby reducing the opportunity for gaining insight.

It is obvious that presenting the data with all their attributes in a single display is not a reasonable approach.

The problem is how to decide what information should be included in visualizations so users can visualize not a bundle of data records but all useful information in the data.

2. What graphical metaphors should be used to represent data?

Since both ATM cells and PDUs are inherently abstract, they do not have obvious graphical representations. Determining which graphical properties should represent the given data attributes is a challenging task. Randomly mapping graphical properties to data attributes will affect:

- the effectiveness of the display, and

  Although any graphical properties can be mapped to any data attributes, not all of the mappings can produce an effective display which supports the tasks users are likely to perform with the data. Inappropriate mappings can even significantly degrade the effectiveness of the display. For example, Carriere [8] stated that using the saturation of the color to indicate the size of files is an inappropriate mapping scheme because files in a Unix system may not be evenly distributed by size. If a large number of files fall into a small range of sizes, the difference in color among those files will be barely distinguished by human vision.

- the ease of understanding.

  If a visualization involves only a very small number of data attributes, it will not be too difficult to understand the graphical representations of the data. However, in ATM protocol data visualization, users always want to

visualize complex relationships among various data items and perform data analysis that involves a large number of data attributes. Integrating multiple data attributes in a single display has already made the display hard to understand. Using inappropriate graphical metaphors not only makes the display more difficult to be understood but also further complicates it.

The problem is inventing appropriate graphical metaphors for representing ATM protocol data so the user can easily distinguish between different data items, spot similar patterns as well as unusual anomalies, and recognize complex relationships among data.

4.1.2 Representations of a Collection of ATM cells

In addition to graphical representations of the data, having an entire information space to be displayed in a single view is also useful for helping users better understand the data. One of the common problems in exploring a large and complex information space is that users easily get lost. This is primarily because the display fails to provide users with some knowledge about the overall structure of the information space. By having a view of the entire information space, the user can see what information is available, how one object is related to other objects, and the location of a specific object. For all these reasons, a view of the underlying information space is crucial in understanding the data.

However, finding a way to efficiently and skillfully display the complete content of the capture buffer in a limited screen space is very difficult. The reasons for this are as follows.

- A huge volume of data must be placed on a small display window.

  In the testing process, up to 8 Mbytes of data, representing 131,271 ATM cells, were collected by the protocol testing system. They were stored in the capture buffer for later analysis. This implies that a huge volume of data needs to be displayed within a small display screen in the analysis process.

- The complete content, not just a small portion of the capture buffer, should be coherently displayed in a single view.

In ATM networks, higher level PDUs are usually transmitted over multiple ATM cells, with these cells distributed over time and interleaved with other services. Users initially have no idea which few data in the capture buffer belong to the same higher level PDU, let alone which portions of the capture buffer to select. A view of the complete buffer provides users with some idea of the data, and how they are related to each other. Therefore, putting the complete contents of the capture buffer in a single view is necessary.

- Placing a large collection of data items within a small display screen may result in the loss of some information.

  The most common problem in Information Visualization is space limitation. The amount of display space available is so limited that only a small amount of information can be visualized in each display. One of the possible solutions for this is to present only the information that is useful for solving the visualization tasks. Such an approach is powerful and expressive, but some information will be lost.

The problem is designing an allocation scheme which makes good use of the display space so the structure of the information space can be maximally apparent to users, and at the same time minimizing the loss of useful information.

4.1.3 Representations of Multiple Protocol Layers

Being able to coherently visualize multiple protocol layers can dramatically reduce the difficulties in understanding the protocol data. Presenting multiple layers of a protocol stack in a single display:

- allows users to easily identify the ATM cells that correspond to the same higher level PDU

  In ATM networks, a single ATM cell can carry only 48 bytes of information. Any unit larger than 48 bytes must be divided into 48 byte chunks and assigned to multiple ATM cells. Typically, a large number of ATM cells are required to transmit a single unit of information in a higher level protocol. Since a single ATM connection can be carrying multiple services, the ATM cells corresponding

to the same higher level protocol data unit are rarely close together; instead, they can be widely distributed over time, with a large number of other cells between them. Therefore, it is very difficult for the user to pick out the ATM cells that belong to the same higher level protocol unit from a huge volume of data.

- can satisfy the users' interest

  Applications seldom use the ATM layer directly to transmit information; instead, they use multiple levels in the protocol stack that are built on top of the ATM layer. At each level of the protocol stack, protocol data units from a lower protocol level are combined to produce a protocol data unit at the next higher protocol level. Typically, many lower level protocol data units are required to produce a single higher level protocol data unit. Since ATM cells for multiple services are interleaved, the protocol data units that are adjacent in time may correspond to different higher level protocol data units. Isolating the protocol data units at a higher level in the protocol stack will not guarantee that all the observed protocol data units belong to the same service. In this case, users may be interested in several protocol levels in order to evaluate whether the information is correctly transmitted.

- helps the user to determine the source of errors.

  Since the lower level protocol data units have no error detection, any error occurring in the payload must be detected at higher levels of the protocol stack. This means that errors are rarely detected at the points where they occur. In order to determine the source of the error, users need to trace from the point of detection to the point where the error first occurs. Due to the fact that a single high level protocol data unit is usually split up into many low level protocol data units and ATM cells, with these cells widely distributed over time and interleaved with many other service data, the point of error detection and the point where the error occurs can be widely separated, both in time and in protocol layers. The tracing process typically involves multiple protocol layers, protocol data units, and ATM cells. Being able to operate at multiple levels of

the protocol stack at once can greatly reduce the time required to locate the source of the error.

However, arranging multiple layers of the protocol stack within a limited display space is extremely difficult. The reasons for this difficulty involve solving the following problems:

- the space limitation,

  In section 4.1.2, we have discussed the difficulties of presenting a large collection of ATM cells in a small display window. Allocating enough space for a single layer of ATM cells is not easy, let alone multiple layers of the protocol stack data.

- the information's visibility,

  Even if there is enough space for allocating all information in a single display, all information presented may not be visible. Since there are too many objects included in a single display, some objects may be occluded by others. This occlusion is a serious problem because it prevents users from gaining insight into the information space and makes it difficult to perceive useful patterns from the display.

- attention switching.

  Even if enough space can be allocated for all information, and all information presented is visible, there is just too much information addressed at once. Whether all information, or only a portion of it, is useful largely depends on the task that users are likely to perform. At times, users may want to perform a task that requires them to focus all their attention on a small portion of information without distraction from other information. At other times, users may need to focus on a particular portion of information in detail while still keeping it in context with other information. In this case, users must be able to divide their attention among different portions of the information.

The problem is devising techniques for further maximizing the use of small screen space so users can coherently visualize not only one layer of data but multiple layers

of the protocol stack data and interactively adjust the degrees of detail at the various levels of the protocol stack.

4.2 Interactions

No matter how adequate representations of the data are, this alone does not automatically solve all problems associated with the visualization of ATM protocol data. Problems such as finding the ATM cells that correspond to the same higher PDU, inspecting particular data, detecting errors and determining the source of error are still unsolved unless the representation scheme incorporates some interaction techniques. With the aid of interaction techniques, users can move forward and backward through time to search for specific features, zoom in a smaller area to examine particular data items, zoom out to get a global view of the information space, and have control over some parameters to modify the presentation of objects.

Although interaction techniques are helpful, they raise a new set of problems when moving in 3D space. With information structures becoming bigger and more complex, it is difficult to organize the entire structure in 2D space. Limited screen space and a huge volume of data encourage the use of 3D metaphors for presenting data. Moving to 3D space solves the clutter problem of 2D displays; however, many of the interaction techniques that have been traditionally used in 2D space must be either modified or replaced with new 3D ones. In order to have an interactive environment for visualizing ATM protocol data, we have to address a few more problems.

1. How to select an object in 3D space.

2. How to interact with our visualization system.

3. How to travel around the information space.

4.2.1 Object Selection

To interact with objects in an information space, the foremost task is to select an object. Object selection typically involves locating the object, then specifying the position and orientation of the object to the renderer. In 2D space, the selection task can be accomplished by two basic operations in 2D direct manipulation systems:

pointing and clicking. The basic idea of this selection scheme is to move the mouse pointer on an object and then perform a clicking action in order to select the object.

However, the same actions become more complicated in 3D space. The major reasons for the complication are:

- the difficulty of performing 3D movements,

  The most commonly available interaction device is the mouse. The mouse is a 2D device. It can be used to directly manage pointer movements in 2D space. In the case of 3D space, pointer movements must be decomposed into series of 1D or 2D movements. For example, in order to move a pointer from (x_i, y_i, z_i) to (x_f, y_f, z_f), the pointer must be initially moved to (x_f, y_f, z_i), and then to (x_f, y_f, z_f). Decomposing 3D movements into a several lower dimensional movements greatly lowers the interaction efficiency.

- the difficulty of reaching for the object,

  Using a mouse-based interaction device, users select an object by moving a pointer on the object. Since the space available in 2D is very limited, the distance between two objects should not be too large. It is not difficult to reach from one object to another. However, there is unlimited space available in a virtual world. Any object can be located at any place in the virtual world. Two objects may be far away from each other; and so, moving from one object to another object may be very time consuming.

- the difficulty of perceiving spatial relationships, especially in depth dimension.

  In 2D space, the user can easily tell whether the pointer is above, below, next to, or on an object. In 3D space, the extra dimension makes object localization difficult, primarily because users have difficulty perceiving spatial relationships among objects within a 3D space, especially in the depth dimension. The pointer which is shown on top of an object may be actually at any distance away from the object.

The problem is developing techniques which facilitate object selection in 3D space so that users can interactively manipulate any object in an information space.

4.2.2 User Interface

In designing a virtual environment for ATM protocol data, it is good to provide users with a set of tools to support the visualization tasks. Tools such as a menu, text window, and potentiometer have been found to be very useful for the following reasons.

- In ATM protocol data visualization, the fundamental objects are those that form the graphical representations of the data. Due to the volume of the data and the complexity of each data item, users may not want to visualize all different data items at the same time. They may only want to examine the data that contribute to their visualization tasks. Therefore, it is necessary to provide users with pull-down menus for selecting the type of the data that they need.
- Although there are many possible ways to arrange different types of information in a virtual world, not all of them can help users better understand the information. The ones that can facilitate the users' understanding depend on what types of information are involved and what tasks users are likely to perform with the information. For example, if users need to compare two series of data items, they may want to put two series in parallel. If two series contain the same type of information, they should not be overlapped with each other. Since the types of information and the tasks that are going to be involved in visualizations are unknown at the implementation stage, it is good to provide users with button boxes or menus for deciding how to present the information.
- Information visualization systems present information graphically. At times, users may want to inspect a particular data item and to get a detailed description of that data. Therefore, it is necessary to provide users with text windows for displaying a textual version of the data.

Although menus, button boxes, and text windows are useful, they were originally designed for 2D space. With the move to 3D space, all these 2D tools can not be directly mapped to 3D ones.

The problem is finding a way to modify the existing 2D tools so they can be used to create a graphical user interface in 3D space, so users can interact with the system through a set of familiar techniques.

4.2.3 Navigation

Navigation is a useful mechanism for overcoming the difficulty in understanding large information structures. The main difficulty with large information structures is that they involve too much information. Even if the display window is large enough to cover an entire information structure, human vision and memory still need to be powerful enough to visualize and memorize everything in the structure at once. Unfortunately, humans can visualize and memorize only a small section of a large information structure at one time. Therefore, comprehending the entire structure at the same time is difficult, particularly as information structures get very large and complex. Being able to navigate in an information space, users can look around the space and find interesting things. By interactively changing the area of interest, users can focus on different portions of an information space and inspect any particular object at will. In this way, the size and complexity of the information space are no longer a problem in understanding the underlying information structure.

Although navigation provides an efficient method for coping with the size and complexity of the information space, navigating in a virtual world induces some difficulties. They are :

1. Users can easily get confused in a virtual world

   In 2D space, users' viewpoints are not a consideration whereas in 3D space, the viewpoint's position and orientation determine the aspects of the visualization that are visible. Therefore, in order to move from one locally visible part of the information space to another, the current position and orientation of the viewpoint must be known beforehand. Getting this information in a virtual world is not easy:

   - User easily lose their overall sense of location while navigating through a virtual world.

Since there is unlimited space available for placing objects in the virtual world, an object may be located anywhere in the world. Searching for a particular object may require users to travel around the entire world. Since users are usually not familiar with the overall virtual environment, traveling in a strange environment may cause them to get lost.

- Users do not gain enough information to identify their current position in the virtual world.

  In visualizations, the fundamental objects are those that represent the data. They look either similar or totally different; however, they are definitely not unique. Therefore, such objects themselves can not be used to identify users' positions.

The problem is designing an environment which not only captures the advantages of 3D space but also simplifies navigation so users can navigate around a virtual world without feeling confused.

2. How to control movements in a virtual world.

   In order to navigate through a virtual world, users must be able to control movements. Just as in driving a car, we can have two basic types of movements: turning left or right, and moving forward or backward. No matter how we drive the car, we notice that the scene is updated continuously. Similar traveling mechanisms and phenomena should be provided by the system so that users can experience a real, 3D space. To achieve this, some problems must be solved.

   - Navigation Mechanisms,

     In order to design a navigation metaphor for our system, two questions are considered: how much freedom users need and how to perform the movement. In 3D space, we can have up to six degrees of freedom. However, is it necessary to provide six degrees of freedom in our visualization environment. The more freedom the system provides, the more difficulty in controlling the movement. In addition to the degree of freedom, the way in which users perform movements should be simple; otherwise, they may lose interest.

- Disruption problem.

  The view must be updated after each user movement. For each update, the graphical hardware re-computes the new coordinates for each object and then renders the result on an output device. Typically, a large number of objects will be involved in the visualization. Therefore, the latency between user movement and change in view may be very long. The long latency results in disruptive animations which lead to confusion for users.

The problem is inventing mechanisms for controlling user movement in a virtual space so that users can travel through the space in a natural way.

Chapter 5

Approaches

In the preceding chapter, we discussed the problems of representing ATM protocol data and interacting with the information presented in 3D space. This chapter will address our visualization approaches in detail.

5.1 Visualizations of Various Types of Data

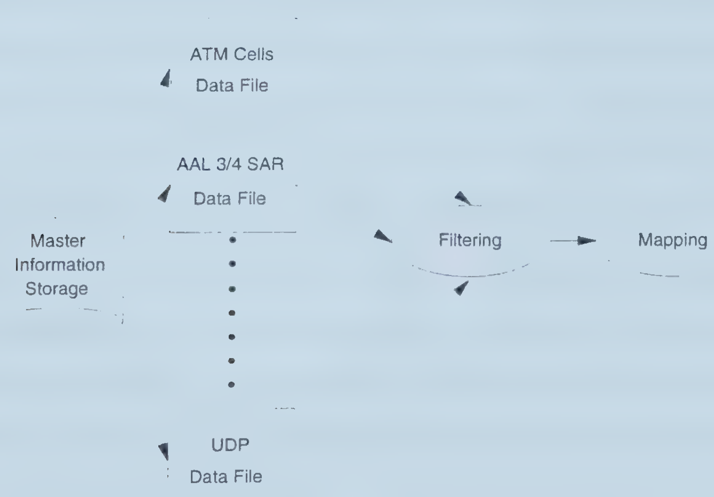


Figure 5.1: Visualizations of a Data Item

The first visualization problem is finding useful and intuitive graphical representations for depicting ATM protocol data. Our approach to overcoming this problem has two major components, which are shown in Figure 5.1: filtering unnecessary attributes and determining graphical mappings.

5.1.1 Filtering Unnecessary Attributes

Filtering out unnecessary attributes is essential to concisely visualize high-dimensional data. High-dimensional data contain a relatively large collection of attributes. For example, the header field of an ATM cell contains at least five attributes. Visualizing the data with all their associated attributes in a single visualization results in a complicated display, low system responsiveness, and users left with the feeling that they cannot handle the flood of information. Being able to filter out unnecessary attributes and displaying only the ones that are useful to users can reduce the space required for the data and the time required for rendering the objects, thereby improving the system's responsiveness, simplifying the overview display, and relieving users' anxiety about the flood of information. The ability to filter unessential attributes is crucial for providing users with a concise visualization of high-dimensional data.

In order to precisely filter out unnecessary attributes, the tasks that users wish to perform with the data must be understood. For example, one of the visualization tasks users would like to perform is analyzing network performance. In ATM cells, not much information is useful for this task. The payload field mainly consists of user service data which may not be helpful in understanding network performance. The header field contains several different attributes. Each attribute performs a different function. Only VPI and VCI in the header are useful for users. Both are used to identify the next ATM switch for the cell and to recognize cells belonging to the same service data unit. They are two pieces of important information in providing appropriate network routing. Therefore, they should be displayed in our visualization. Whether a data attribute should be filtered out depends on the tasks users are likely to perform with the data.

5.1.2 Determining Graphical Mappings

Graphical presentations of data provides users with an expressive and effective display of the data. A useful overview display not only provides users with the overall structure of the information space but also brings out the content of the space and even highlights significant data. With this display, users no longer worry about coping with a huge information space. They can just look at the display and decide which

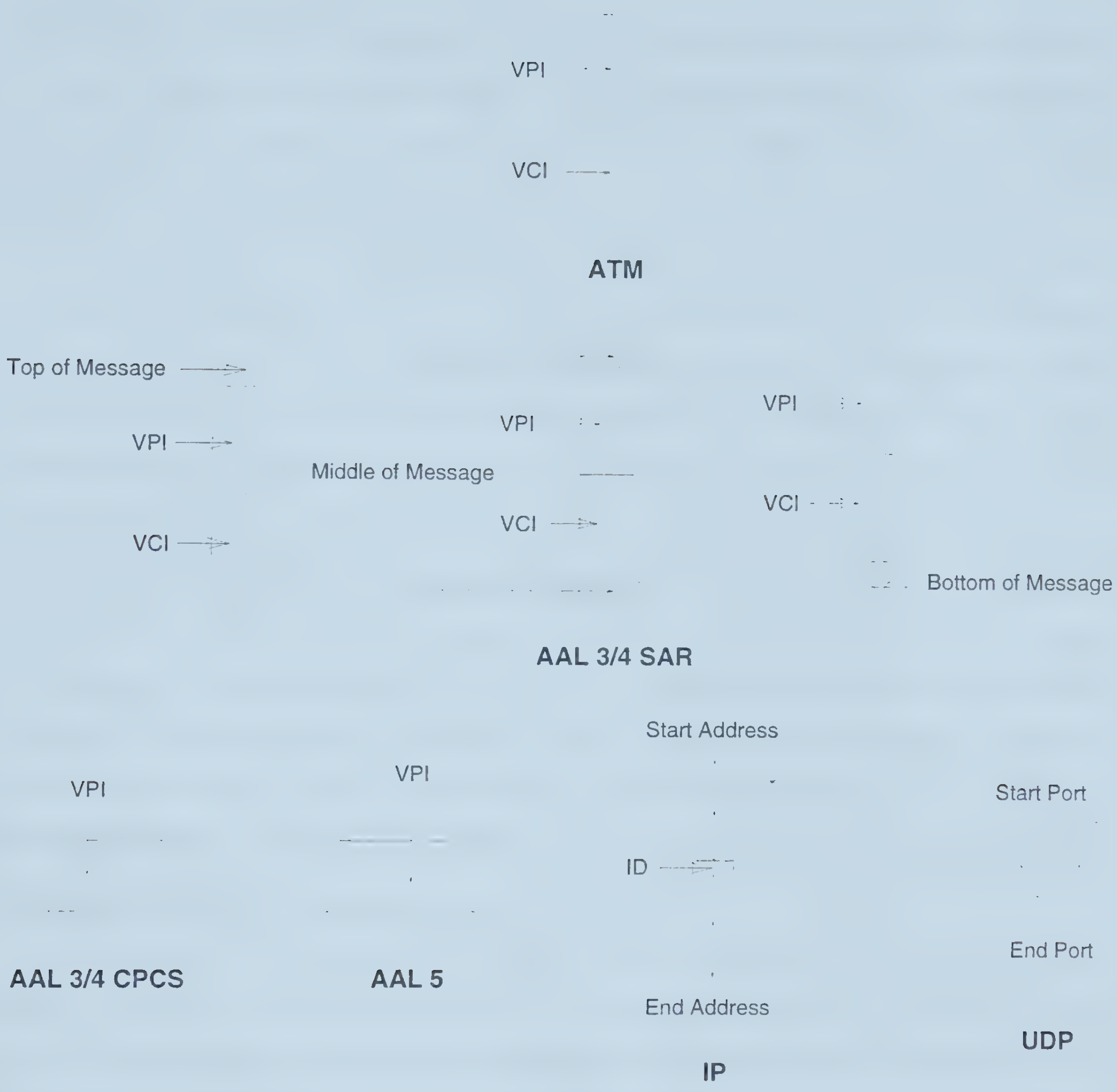


Figure 5.2: Graphical Encoding of Different Cells

part is useful. To achieve this purpose, graphical properties including color, shape, size, and texture are used to encode the content of the data. That means that instead of textual or numerical presentations, graphical views are displayed in visualizations. Each of the graphical properties in the display represents and reflects some combination of the semantic values in the original data. For example, figure 5.2 shows how different cells are represented by different shapes. Due to the fact that humans can easily recognize features, recall similar images, and spot exceptions in graphical displays [3], presenting data graphically is an effective method for understanding data with multiple attributes and their relationships.

While graphical representations of data are helpful in understanding the information space, defining mappings between the data attributes and the graphical properties is crucial. Indeed, the mapping and the data in the underlying information space control the actual appearance of the view, thereby affecting the effectiveness of the visualization. For example, if different attributes are mapped to different shapes, the shape of the items in the view would be controlled by the type of the attribute the item represents, and how much information users can perceive depends on how concise and lucid this overview display is. Therefore, it is necessary to have appropriate mappings between graphical properties and data attributes. Since the number of graphical properties that can be used to depict attributes is usually less than the total number of attributes in the underlying information space, only the important attributes should be mapped to the graphical properties. In order to have appropriate mappings, we must consider not only the graphical properties that are available but also the attributes that are useful to users. Deciding which attributes are useful and which attributes should be filtered out is discussed in section 5.1.1. The question remaining is how to define the mappings which generate a concise and lucid presentation of information.

In [20], Mittal et al stated, "One factor that determines the difficulty of understanding encoding techniques is the number of dimensions involved." This means that the more complicated the encoding techniques, the more difficult for users to understand the attributes shown. Therefore, simple but effective mappings are used in our

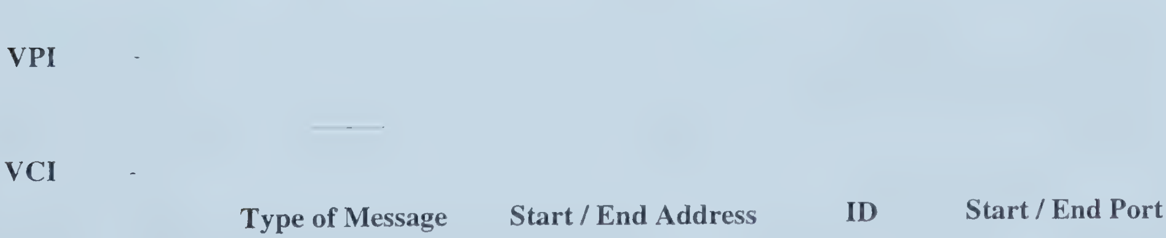


Figure 5.3: Graphical Objects Appeared in Visualizations.

visualization. Figure 5.3 shows mappings of different attributes. In this example, different shapes are used to represent the various attributes; the color of the object is mapped to the value of the attribute. Figure 5.2 shows representations of different cells which are the combination of different attributes.

5.2 Visualization of a Collection of ATM Cells

The second problem is how to allocate a large collection of cells on a single small screen without causing any visual clutter. Several possible solutions to this problem have been proposed:

- using tree structure to represent hierarchical information. [27].
- using nested rectangular boxes to represent hierarchical information [26].
- presenting information on a perspective wall in 3D space [19].
- placing the nodes on a globe and drawing arcs between them to encode their linkage [9].

Although all of these techniques make use of different geometries to depict the basic structure of the visualization, they share the same idea: presenting information in a three dimensional space. Using three dimensional visualizations seems to be the most promising solution to the space limitation problem. The reasons for this are:

- There is more space available in 3D than 2D so the designer can have more freedom in the placement of information.
- The extra degree of freedom in 3D space allows more elements to be placed in the visualization environment, so two or more relationships can be represented.

In our visualization, we choose a ring structure as a background. The next subsection shows how a large collection of cells can be allocated on the ring structure and why we chose this structure for our visualization.

5.2.1 Ring Information Layout

Our visualization is based on a ring structure. The ring can be viewed as a short cylinder, with the complete contents of the capture buffer being mapped onto the inside surface of the cylinder. Users are initially at the center of the cylinder and looking towards the point that represents the start of the capture buffer. Users can get a complete view of the capture buffer by rotating about the vertical axis at the

center of the cylinder. There is a small gap in the cylinder between the start and end of the capture buffer so that users can easily identify the starting and ending position of the ring.

In order to precisely lay out a collection of cells on a ring structure, the size of the ring must be determined. Then all related and important information can be positioned on and around the ring.

5.2.1.1 Information On the Ring

One of the important properties associated with a cell is time. Usually this time is the time when the cell was received by the test equipment. In some cases, the absolute time when the cell received is important, and in other cases the time difference between cell arrivals is important.

5.2.1.2 Information Around the Ring

After the ring is set up, the next step is to allocate a sequence of cells around the ring. How do we distribute a collection of cells around the ring? There are two types of distribution: regular and irregular.

1. Regularly Distributed Cells

   For regularly distributed cells, the allocation is straight forward. Each data set should have an attribute which indicates the arrival time of the cell. This time is used as a starting point on the ring. The width of the cell depends on the size of the cell and the bandwidth of the ATM connection. For example, an ATM cell consists of 53 bytes.

2. Irregularly Distributed Cells

   Using a regular algorithm to allocate irregularly distributed cells is inappropriate. The main purpose of distortion based visualization techniques is to increase the amount of information that can be displayed on the screen by distorting the coordinate system used to display the information. This distortion is based on allocating more screen space to the parts of the visualization that are more important, and less screen space to the parts that provide contextual information. These techniques are based on identifying hot spots in the visualization.

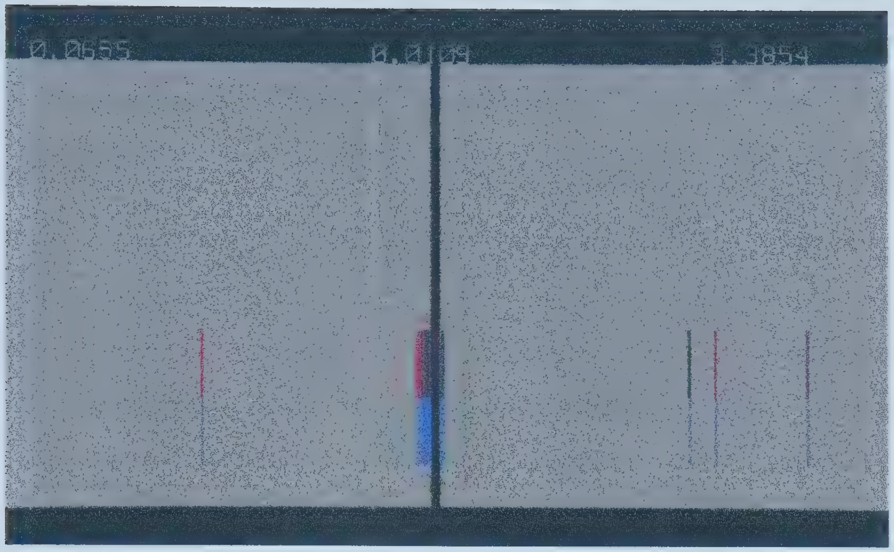


Plate 5.1: A ring with no compression.

These hot spots are the aspects of the visualization that hold the user's current interest. The rest of the visualization provides the context for the hot spot. Although users are not interested in the details of the contextual information, it provides the framework in which the information in the hot spots is interpreted. A distortion can be viewed as an expansion and contraction of the information space. Near the hot spots the information space is stretched to give more screen space to users' points of interest, and outside of the hot spots the information space is compressed. Thus, more detail can be displayed at the hot spots. The different distortion techniques vary in the way that they treat the hot spots and the nature of the distortion.

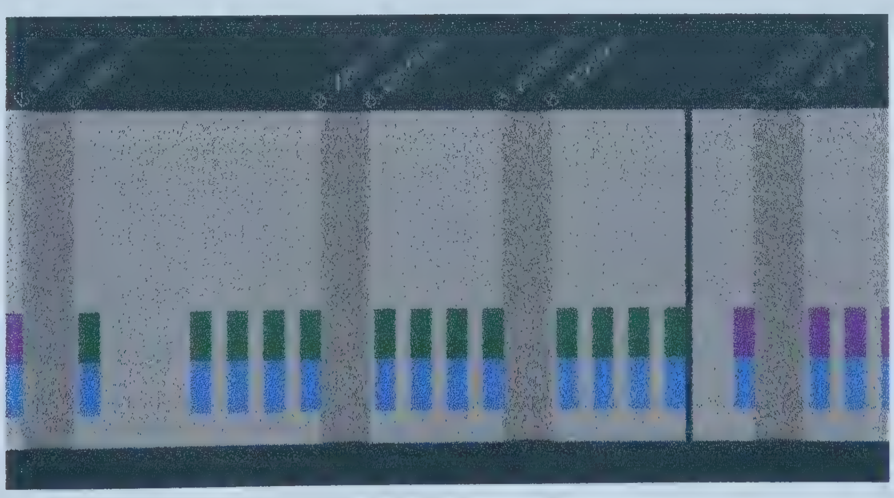
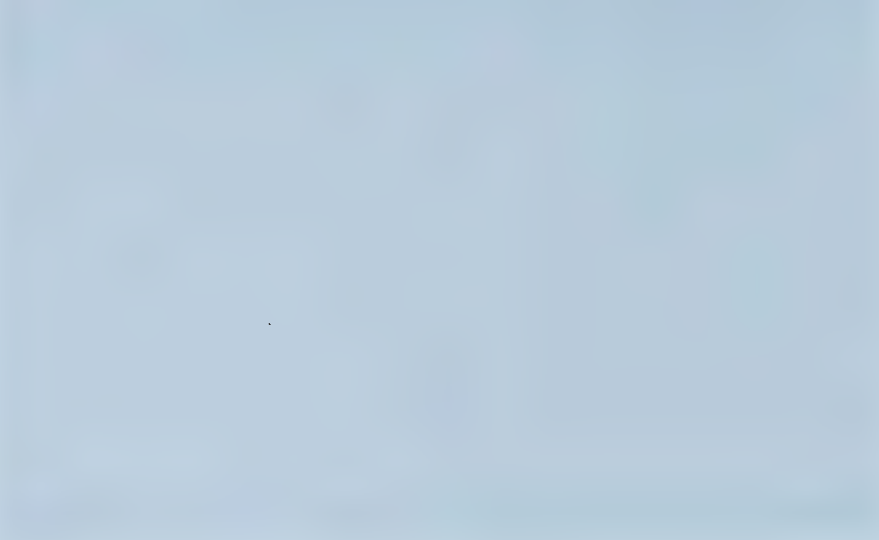
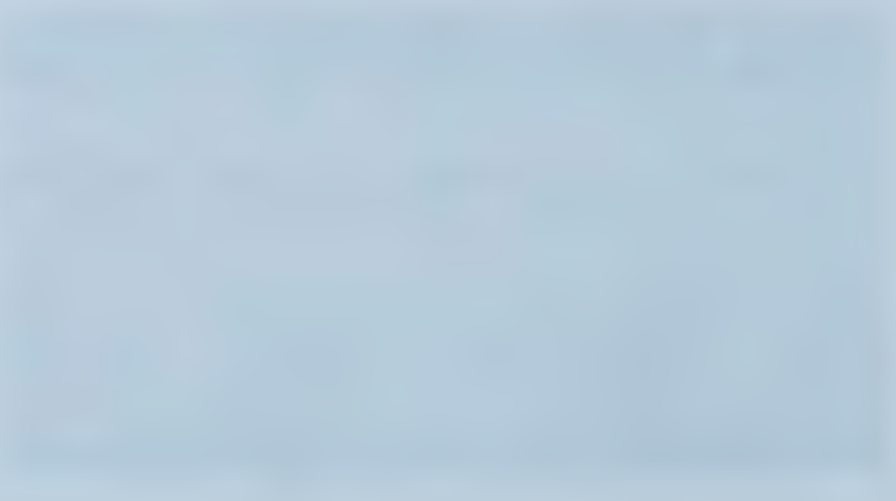


Plate 5.2: A compressed ring.





5.2.2 Displaying Global and Local Views

Integrating both global and local views of an information space is helpful in analyzing the underlying information. Filtering unnecessary attributes and applying graphical representations of data can reduce the amount of space required for depicting each data item, increasing the amount of information that can coherently be displayed on the screen and providing users with a global view of the information space. One of the major problems with this global view is that some details about the data would be lost to the user. Focus+Context techniques [10, 19] provide users with an ability to visualize an entire information structure and focus at specific items simultaneously. With these techniques, users can search through the entire information space for any interesting pattern and examine useful details of specific items without losing context.

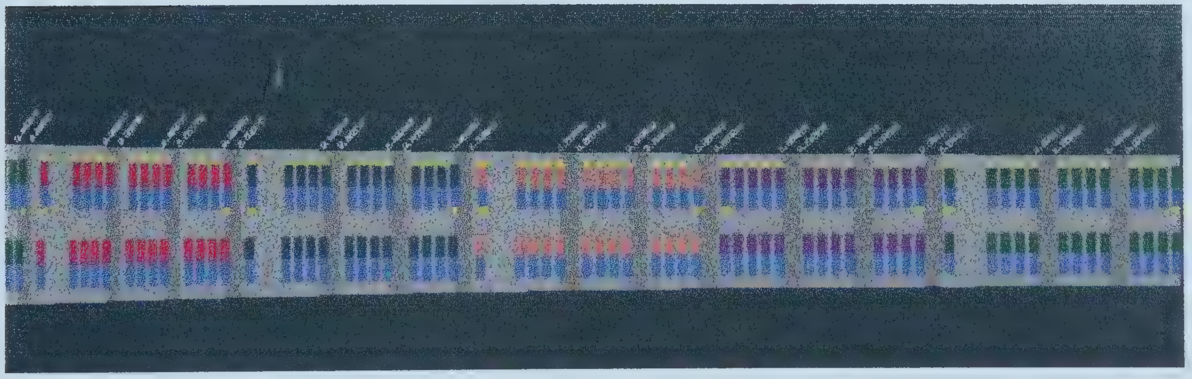


Plate 5.3: Global View

A similar idea is provided in our visualization. Initially, users will be shown the global view of the information space as in Figure 5.3. When users need the details of a particular cell, those details will be displayed on a three dimensional panel. Figure 5.4 shows a view in which the user requested to see the details of a particular AAL 3/4 SAR cell. (Note that three dimensional renderings are used to highlight the selected cell and the panel at the top right corner is used to show the details of the cell.) This view shows that both global and local views of the information space are integrated on a single display.

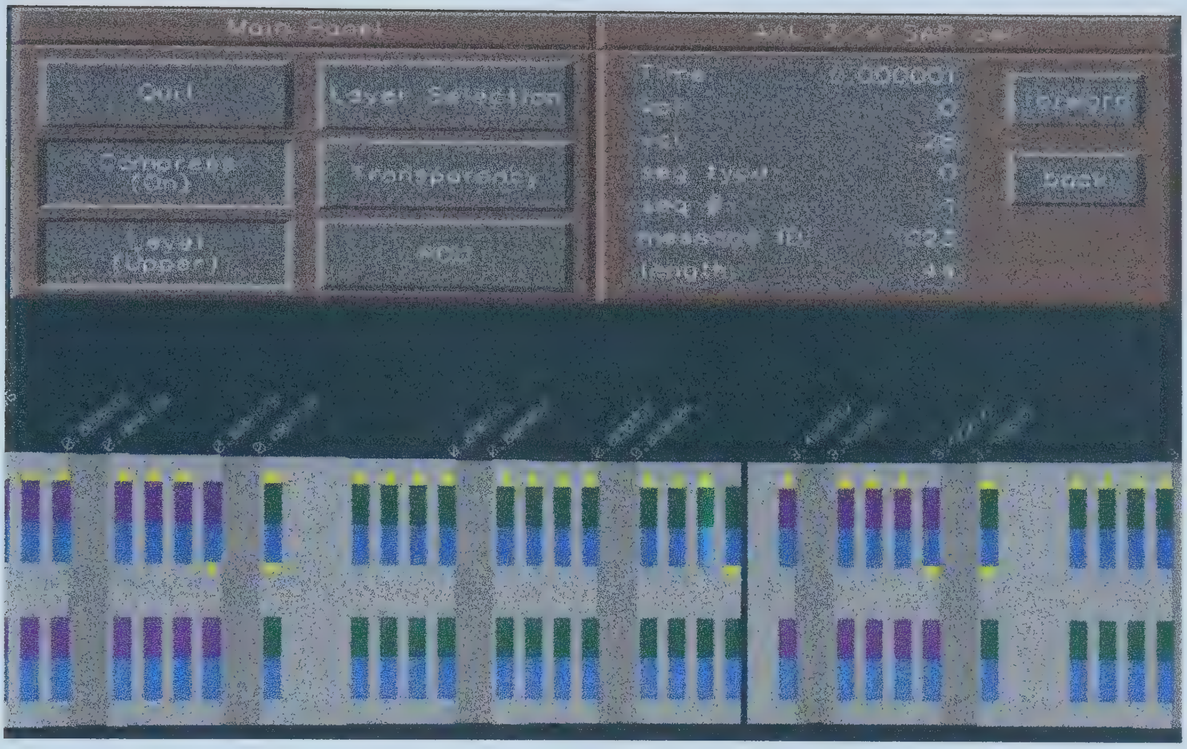


Plate 5.4: Global + Local Views

5.3 Visualization of Multiple Protocol Layers

The third problem is how to coherently visualize several layers of information.

5.3.1 Multiple Layers of Rings

One possible solution to this problem is to further maximize the use of display space by stacking multiple rings one on top of the other. The 3D visualization approach that we have discussed in section 5.2 utilizes display space by placing a layer of information around a 3D ring. Such a design maximizes the efficient use of the small screen space; however, it is still not good enough to solve the problem associated with visualizing multiple layers of information. To coherently present multiple layers of information, multiple ring structures must be included in a single display. One possible way to achieve this is to stack rings one on top of the other in a vertical or horizontal direction. By stacking multiple rings in two directions, more layers of information can be included in a single display, thereby further maximizing the use of display space in visualizations.

5.3.2 Transparency

Stacking multiple rings one on top of the other allows more layers of information to be included in a single display, but, doing so does not mean that all information included can be visualized by users. The objects at the back may be blocked by the objects in the front. In order to make all objects become visible, a special technique must be used.

Transparency, one of the more powerful techniques that has become practical in 3D graphics, can be used to solve this occlusion problem. With normal rendering, background objects may be obscured by opaque foreground objects. Through the use of transparent rendering, a semi-transparent object only partially obscures the objects behind it; therefore, all background objects are visible through foreground objects.

The Use of Transparency in ATM Protocol Data Visualization :

Transparency can be used to show different layers of the protocol stack. In this visualization, the ATM cells form the background and are drawn on the ring with the largest radius. The next layer of the protocol stack is drawn on a ring with a slightly smaller radius, and is semi-transparent. Thus, users can see both the ATM cells and the PDU's that are constructed from them. This gives a good view of the time relationships between the cells and the PDU's and the context in which the PDU's are assembled. This process can be continued with multiple layers of the protocol stack to show how several layers are related.

The Advantage of Using Transparency :

The main advantage of using transparency is that it allows several representations to be compared without the use of extra screen space and some mechanism to show how the representations are related. For example, two streams can be drawn on top of each other, with the front most stream drawn in a semi-transparent way. With this visualization users can easily see how the two streams are related to each other since they are both using the same time scale and are superimposed on top of each other. Similarly, several layers of a protocol stack can be displayed on top of each other. In

this way, a high level PDU and the lower level PDUs it is composed of can be seen at the same time, along with the time relationships between them. By controlling the transparency of the different layers, users can focus on different parts of the protocol stack and easily move back and forth between them without losing context.

5.4 Object Selection

Many visualization systems present data in 2D space where it is not difficult to manipulate the displayed information. The actions of pointing and dragging are two basic operations in 2D direct manipulation systems. However, the same ideas cannot be mapped directly to our three dimensional visualization.

5.4.1 3D Interaction Device

The interaction devices for our system are a pair of trackers. A tracker is a six-dimensional device in which up to six degrees of freedom is provided. Each tracker is equipped with a tracking sensor and has three buttons attached to its the surface. The tracking sensor is used to sense the position and orientation of the tracker in 3D space. The tracker buttons are used to invoke operations such as selection, translation. and rotation.

With the use of trackers, users can not only directly but also interactively manipulate objects in 3D space. Since the tracker can provide the renderer with six-dimensional information, 3D movements no longer need to be decomposed into several lower dimensional movements. Users can directly interact with any object in 3D space. More importantly, the tracker can provide the system with a very acceptable update rate; and therefore, it supports a certain level of responsiveness in the environment. All users' movements, including performing gestures and activating operations, can produce responses within a reasonable time.

5.4.2 Laser Gun

One of the popular methods to select objects in a virtual world is a laser gun [17]. The main idea is to shoot a ray from the tracker along its x-axis into the scene. The

first intersected object in the scene will be the one that can be selected. By simply activating one of the tracker buttons, the object is selected.

The laser gun method provides users with a more efficient way of reaching an object than the traditional pointing method, which requires explicit contact of the pointer with the object. In small virtual environments, this explicit contact approach may not cause any difficulties in reaching an object because objects are not widely separated. However, the virtual space required for ATM protocol data is always very large. The explicit contact approach makes object selection inefficient. Consider the following example. Figure 5.4 shows that there are three named objects and a pointer

Figure 5.4: Object Selections

in a 3D space. Suppose a user initially selects object 1, then 2, and finally 3. By using the traditional pointing method, the user needs to move the pointer from the current location to the point at which the object is located. This task can be accomplished by moving the mouse with the arm across the mouse pad. The dotted lines in Figure 5.4 show the shortest path required for reaching all objects in the specified order. By using laser gun method, the user simply rotates the tracker with the fingers to adjust the orientation of the tracker, so the ray shooting form the tracker into the scene can interact with the desired object. The four lines in Figure 5.4 represent the initial ray r_0 and the three rays r_x which intersect with object x where x is 1, 2, or 3. The angle between r_i and r_{i+1}, where i varies from 0 to 2, is the amount of change needed in the orientation of the tracker. With the laser gun method, users do not need to explicitly contact the object; instead, they point at the object from some distance

away. Therefore, it is more efficient to select an object by using a laser gun than the traditional pointing method.

5.5 User Interface

Although many user interface techniques such as menus, button boxes, text windows, and graphical potentiometers were originally designed for 2D space, the tasks they perform can still be applied in 3D space. It is better to keep using the existing 2D techniques in 3D space because users do not need to spend time learning new techniques. The MR Panel Package, which is one of the packages supported by the MR Toolkit [11], provides a simple way of implementing 2D user interface techniques in a 3D space. In this section, we focus on how 2D user interface techniques are modified in the panel package so that they can be used in 3D space.

5.5.1 MR Panels

To program a user interface, the programmer needs to decide where the interface is located, what interface techniques are included, and what task each technique performs. Whether the space is 2D or 3D, the types of interface techniques required and the tasks they perform don't change. They depend mainly on the types of interactions that users are likely to perform. However, the location of an interface is greatly affected by the dimensions of the space. Extra dimensions imply that extra information is required to define the location of the interface.

In the MR panel package, an MR panel stimulates a 2D screen to produce a display area for presenting a set of standard 2D interface techniques such as menus, button boxes, text windows, and graphical potentiometers. Whether the space is 2D or 3D, a display screen is required to display the interface techniques. In 2D space, a display screen is defined by its 2D position, orientation, and size. In 3D space, the definition looks very similar, except that 2D position and orientation are replaced by 3D ones. An MR panel is a flat rectangle in 3D space.

5.5.2 MR Panel Selections

In section 5.4, we discussed how an object can be selected in 3D space. The panel package makes use of a similar approach for selecting an interaction technique in a 3D panel. The user holds a tracker with one hand. A ray is emitted from the tracker into the 3D space of the application. The first front-facing panel intersected by the ray is the panel selected. Within the selected panel, the point where the ray intersects with the panel determines the active 2D interaction technique. Through the activation of one of the tracker buttons, the active 2D interaction technique is selected.

5.6 Navigation

The two common problems associated with navigating in a virtual 3D space are that users easily get lost and they find it hard to control their movements. In order to solve these problems, it is useful to provide users with some cues for recognizing the current position of the viewpoint and to have familiar mechanisms for navigating around the space. In this section, we discuss how we facilitate position recognition and movement controls in our visualizations.

5.6.1 Position Recognition

Traveling through a virtual environment causes users to quickly become disoriented. The main reason is that users do not have the same experience they have in the real world. For example, when we are traveling in a new city, how do we know our current location? We can look at the address of a nearby building. Similarly, when users are navigating around a virtual world, they may want to know where they are, and how they can go to specific locations. They may expect that information about their current position can be found in the virtual world. In our visualizations, coupling the structure of the information space with the time marked around the ring can help users to recognize the current position of the viewpoint.

5.6.1.1 Structure of Information Space

The ring structure of the information space can prevent users from losing their overall sense of location. In our visualizations, all information is displayed around the inner

surface of the ring. To examine a particular region of the ring, users simply restrict the view to that region. No matter where the region is, the viewpoint is still kept somewhere inside the ring. This means that instead of the entire virtual world, only the area bounded by the ring is involved in navigation. Since users are accustomed to a ring structure, navigating through the ring helps them maintain overall context. Thus, the possibility of users' getting disoriented in the virtual world is reduced.

5.6.1.2 Time

The time marked around the ring can provide users with perceptual information about the viewpoint position. The numbers around the ring are the times at which data items are collected by the capture buffer. They also serve as addresses in our 3D environment. Restricting the display to a ring allows users to restrict their navigation inside the ring. By looking at the numbers marked around the ring, users can know where they are inside the ring and where a particular location is. For example, figure

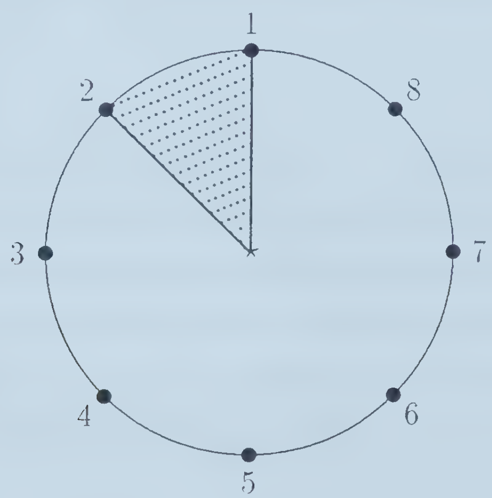


Figure 5.5: Position Recognitions

5.5 shows a marked ring with a point and a shaded region inside. The point and shaded region represents the user's viewpoint and view respectively. The numbers around the ring indicate the addresses of the ring. Suppose the user wants to inspect information at address number 5. He knows he is initially around address number 1. He needs to rotate the viewpoint until he sees address number 5. With this kind of setup, the user can easily perceive where he is and how he can visit any other position

within the ring.

5.6.2 Movement Control

Being able to control the position of the viewpoint, users can determine the portion of the information space that is displayed. However, traveling in a virtual world is not as easy as in the real world. In section 2, we discussed the difficulties with movement controls in 3D virtual worlds. In the following, we explain how each of these difficulties is solved in our visualizations.

5.6.2.1 Walking

The navigation metaphor that we used is "walking". The main characteristic of this metaphor is that the user's viewpoint is always restricted to the ground floor. To inspect a particular object, the user utilizes the tracker to adjust the view's position and direction so that the view is facing to the object. Two types of movements provided in our system are rotation and translation.

1. Rotation

   Rotation enables users to adjust the direction of the view. By rotating the view about a vertical axis, users can look around the entire information space. To perform rotation motions, users simply activate one of the tracker buttons to toggle to a rotation mode and then rotate the tracker with the fingers. How much the tracker rotates determines how much the view will be changed in the virtual world. In this way, users can interactively adjust the direction of the view.

2. Translation

   Translation enables users to move the viewpoint forward and backward along the line of sight in a virtual world. By moving the viewpoint closer to the ring, users actually restrict their view to a smaller portion of the information space and visualize a detailed presentation of the information in that region. By moving the viewpoint away from the ring, users expand their view so that they can perceive more information at one time. To support zooming in and

out, two tracker buttons are used. One is for zoom in and the other one is for zoom out. Users simply activate the zoom in/out button to move the viewpoint forward/backward. How much the viewpoint moves depends on how many times the button is activated.

5.6.2.2 Selective Rendering

As we design a representation scheme for ATM protocol data, we have considered the responsiveness of the system. Filtering out unnecessary data attributes and using simple imagery, both of which have been discussed in section 5.1, are two strategies for improving the responsiveness of the system. In order to further improve the system performance, we employ a selective rendering approach. The main idea behind this approach is to render the objects that are within the field of view in the display window. This means that only the objects that are visible to users will be rendered.

Using the selective rendering approach can substantially improve the responsiveness of the system. The latency between user movement and change in view highly depends on the number of objects involved in the rendering process. The more objects that need rendering, the longer the latency will be. The selective rendering approach minimizes the number of objects involved in each display update in order to lower the time required for scene rendering, thereby maintaining a certain level of responsiveness in the environment.

Chapter 6

Implementation

After presenting our approaches to the problems of ATM protocol data visualization, we describe how these approaches can be implemented and how our prototype system is used for creating a real time visualization of ATM protocol data.

6.1 Implementation Strategy

Generating a graphical view of ATM protocol data requires us to:

- determine the size of the ring,
- label the ring,
- position the cells,
- update the location of the viewpoint after the view has been changed, and
- render a transparent layer.

This section will present the strategies that we used to solve each of the above questions.

6.1.1 Computing the Ring Size

The size of the ring affects the efficiency of the visualization. A ring acts as a background in our visualization, where the complete content of the capture buffer is mapped on its inside surface. A ring of vast size provides enough room for allocating the elements but has the drawback that its surface is too far away from its center. Therefore, details are initially too small to be seen, and it takes a lot of steps to go

from the center to the surface of the ring. Alternatively, a tiny ring has a smaller radius, but does not have enough space for placing all required elements. The display becomes cluttered and jumbled because of element overlapping. Choosing a proper size for the ring is crucial in developing an efficient visualization. The question is how to determine the size of the ring.

We devised a formula to calculate the radius of the ring. If the capture buffer is considered as a rectangular sheet, the time difference between the first and last cells collected will determine the length of the sheet. If the sheet is rolled up into a cylinder or ring, the circumference of the ring will be the same as the length of the sheet. This means :

$$\text{timeDiff} = 2\pi R$$

where timeDiff is the time difference between the first and last cells collected in the capture buffer and R is the radius of the ring.
The radius of the ring is then calculated with:

$$R = \text{timeDiff} \times c_2 + c_1 \tag{6.1}$$

where both c_1 and c_2 are constants. c_1 is used to guarantee that a certain size of radius is used. c_2 is used to scale the value of timeDiff.

6.1.2 Computing TimeLine

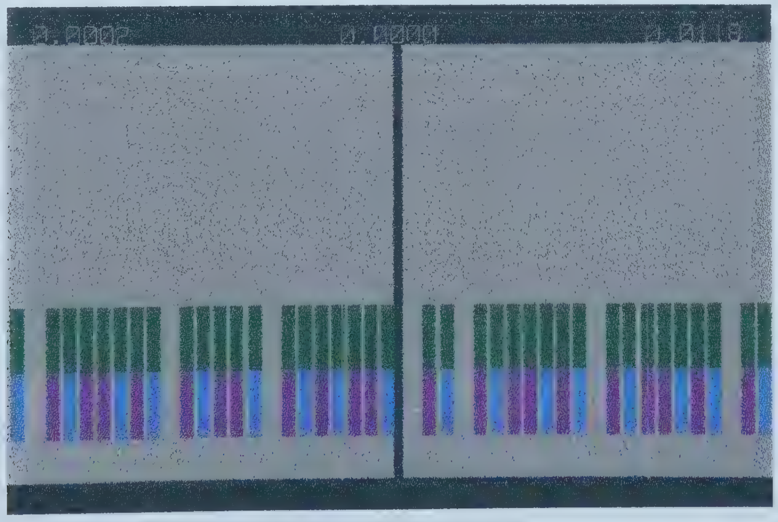


Plate 6.1: A Labelled Ring.

After the ring has been determined, the second task in our implementation is to label it. Figure 6.1 shows a labelled ring. The numbers at the top of the ring indicate the times at the start, the current point and the end of the buffer. The question is how to determine the positions of those numbers.

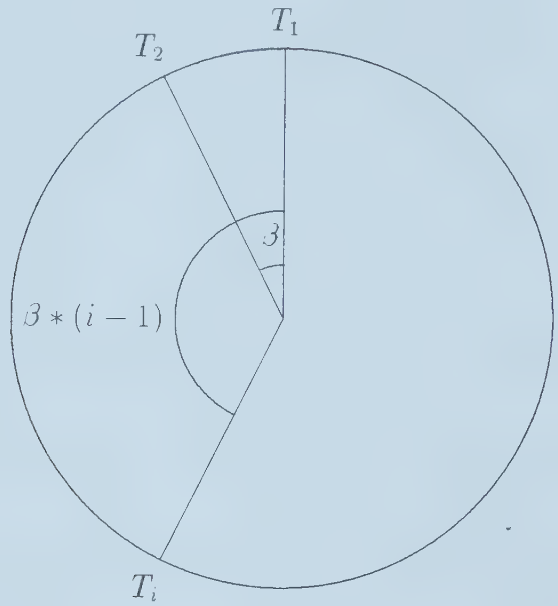


Figure 6.1: Compute Time on the Ring

If a ring is equally divided into n sectors, as shown in figure 6.1, each sector will be at an angle of β radian where $\beta = \frac{2\pi}{n}$. The first sector starts at zero, and the second sector is β away from zero. Therefore, the i^{th} sector is $\beta * (i-1)$ away from zero.

The time at the i^{th} sector, T_i, is defined by:

$$\frac{\beta * (i-1)}{2\pi} * \text{timeDiff}, \tag{6.2}$$

and, the number T_i will be positioned at:

$$Location\ of\ T_i = \begin{pmatrix} R * \cos[\beta * (i-1)] \\ R * \sin[\beta * (i-1)] \\ height\ of\ the\ ring \end{pmatrix}. \tag{6.3}$$

6.1.3 Computing Object Position

The final task in our implementation is to allocate graphical objects around the inside surface of the ring. The position and size of an object are determined by a cell's or

PDU's arrival times and the time interval occupied by the cell or PDU in the capture buffer respectively. Figure 6.2 shows one of the objects found on the ring which represents an ATM cell. The v_1, v_2, v_3, and v_4 are the four corners of the object, and the h is the height of the object.

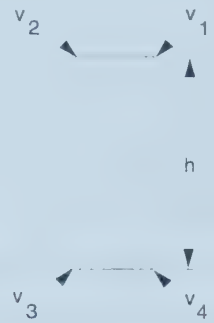


Figure 6.2: An object represents a ATM cell.

Assume the cell arrived at time t. Therefore, v_1 *and* v_4 are defined as:

$$v_1 = \begin{pmatrix} R * \cos\alpha \\ R * \sin\alpha \\ z + h \end{pmatrix}, \quad v_4 = \begin{pmatrix} R * \cos\alpha \\ R * \sin\alpha \\ z \end{pmatrix}. \tag{6.4}$$

where $\alpha = \frac{t*2\pi}{\text{timeDiff}}$, and z is the base of the object.
If the size of the cell is $size$ bytes, and the number of bytes transmitted per unit time is $freq$, the maximum time interval occupied by the cell will be $\frac{size}{freq}$. Therefore, the v_2 *and* v_3 of the object are:

$$v_2 = \begin{pmatrix} v_1 + \Delta x \\ v_1 + \Delta y \\ z + h \end{pmatrix}, \quad v_3 = \begin{pmatrix} v_1 + \Delta x \\ v_1 + \Delta y \\ z \end{pmatrix}. \tag{6.5}$$

where

$$\begin{aligned} \Delta x &= R * (\cos\theta - \cos\alpha), \\ \Delta y &= R * (\sin\theta - \sin\alpha), \; and \\ \theta &= \frac{size * 2\pi}{\text{timeDiff} * freq}. \end{aligned}$$

6.1.4 Computing Viewpoint Position

The most important task in our implementation is to compute the current viewpoint position. The viewpoint position indicates the point where users are looking from in

a 3D virtual world, and it can help to specify what part of the information space is visible. Therefore, it is an important piece of information in our visualizations.

Computing the current viewpoint position is a two step process. First, we determine a change of eye position. Then, based on this change, the viewpoint position is updated.

1. Determining a change of eye position

   This requires us to detect the eye position before and after the user's movement. We use a tracking device to locate the eye position. The tracking device provides the location and orientation of any object to which it is attached. Quaternions are used to represent orientation and rotations. In our study, a tracker is held in one hand, and the user's viewpoint in the 3D virtual world is positioned at the tracker. Users simply manipulate the tracker with his fingers to adjust the viewpoint position, and the tracking device records the viewpoint position.

   Suppose the eye position before and after movement are eye_{before} and eye_{after} respectively, where

$$eye_{before} = \begin{cases} \begin{pmatrix} ex_{before} \\ ey_{before} \\ ez_{before} \end{pmatrix} & \text{eye location before movement;} \\ \begin{pmatrix} e\omega_{before} \\ ea_{before} \\ eb_{before} \\ ec_{before} \end{pmatrix} & \text{eye orientation before movement.} \end{cases}$$

$$eye_{after} = \begin{cases} \begin{pmatrix} ex_{after} \\ ey_{after} \\ ez_{after} \end{pmatrix} & \text{eye location after movement;} \\ \begin{pmatrix} e\omega_{after} \\ ea_{after} \\ eb_{after} \\ ec_{after} \end{pmatrix} & \text{eye orientation after movement.} \end{cases}$$

   Therefore, the change of eye position is:

$$change = \begin{cases} \Delta location & \text{a change of eye location;} \\ \Delta orientation & \text{a change of eye orientation.} \end{cases}$$

where

$$\Delta location = \begin{pmatrix} ex_{after} \\ ey_{after} \\ ez_{after} \end{pmatrix} - \begin{pmatrix} ex_{before} \\ ey_{before} \\ ez_{before} \end{pmatrix}; \tag{6.6}$$

$$\Delta orientation = \begin{pmatrix} e\omega_{before} \\ ea_{before} \\ eb_{before} \\ ec_{before} \end{pmatrix}^{-1} \begin{pmatrix} e\omega_{after} \\ ea_{after} \\ eb_{after} \\ ec_{after} \end{pmatrix}. \tag{6.7}$$

2. Updating the viewpoint position.

   After a change of eye position has been found, the current viewpoint position can be computed. Suppose the viewpoint is initially located at $view_{before}$ where

$$view_{before} = \begin{cases} \begin{pmatrix} vx_{before} \\ vy_{before} \\ vz_{before} \end{pmatrix} & \text{viewpoint location before change;} \\ \begin{pmatrix} v\omega_{before} \\ va_{before} \\ vb_{before} \\ vc_{bafore} \end{pmatrix} & \text{viewpoint orientation before change.} \end{cases}$$

   Therefore, the viewpoint after user's movement, $view_{after}$ is definded as:

$$view_{after} = \begin{cases} \begin{pmatrix} vx_{before} \\ vy_{before} \\ vz_{before} \end{pmatrix} + \Delta location & \text{view location after change;} \\ \begin{pmatrix} v\omega_{before} \\ va_{before} \\ vb_{before} \\ vc_{bafore} \end{pmatrix} \Delta orientation & \text{view orientation after change.} \end{cases} \tag{6.8}$$

6.1.5 Computing Visible Portion

After determining the current viewpoint position, we can compute the visible portion of the information space. Because a large number of objects are involved in the information space, displaying all of them results in increasing the time required for scene rendering. Therefore, we render only the objects which are within the user's view. To do this, we have to determine the user's visible portion.

Assume the field of view is $2 \times \theta$. Figure 6.3* shows the visible region of the information space. *viewpoint* is the user's view position. It can be anywhere in the

*Here, only x and y coordinates are used because the ring is on x-y plane.

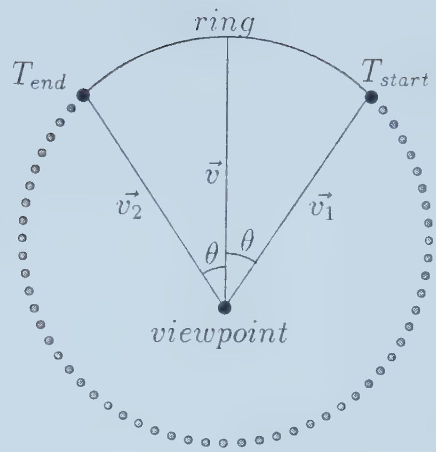


Figure 6.3: A Visible Portion.

virtual world. $\vec{v}$ is the direction at which the eye is looking. Both $\vec{v_1}$ and $\vec{v_2}$ are θ away from $\vec{v}$. This means that the region bounded by $\vec{v_1}$, $\vec{v_2}$ is within the user's view. In our visualization environment, all graphical objects are mapped around the inside surface of the ring. Therefore, the *ring*, $\vec{v_1}$, and $\vec{v_2}$ form a closed region. Only the objects which are inside the closed region are visible to users. Determining this closed region requires us to compute three control points which are *viewpont*. T_{start}, and T_{end}. The algorithm for computing the viewpoint position has been discussed in section 6.1.4. The question left is how to compute T_{start} and T_{end}.

Determining T_{start} and T_{end} can be viewed as ray intersection problems. $\vec{v_1}$ and $\vec{v_2}$ are the rotations of $\vec{v}$ by θ and $-\theta$ degrees about the z-axis. The directions of both v_1 and v_2 and the *viewpoint* define two rays: ray_1 and ray_2. T_{start} and T_{end} are simply the intersections of these rays with the *ring*. Since the *ring* is a circle on xy plane, the problem of locating T_{start} and T_{end} can be solved by determining the intersection points of two rays with a circle.

Let

$$\vec{v_1} = \begin{pmatrix} v1_x \\ v1_y \end{pmatrix}, \quad \vec{v_2} = \begin{pmatrix} v2_x \\ v2_y \end{pmatrix};$$

and the user's view is located at:

$$viewpoint = \begin{pmatrix} view_x \\ view_y \end{pmatrix}.$$

The equations of the two rays are defined as:

$$ray_1 = \begin{pmatrix} view_x + d_1 * v1_x \\ view_y + d_1 * v1_y \end{pmatrix}, \quad and \quad ray_2 = \begin{pmatrix} view_x + d_2 * v2_x \\ view_y + d_2 * v2_y \end{pmatrix}$$

where both d_1 and d_2 are constants.

The equation of the *ring* is:

$$x^2 + y^2 = R^2$$

where R is the radius.

Solving for the point where ray_1 intersects with $ring$ for d_1

$$(view_x + d_1 * v1_x)^2 + (view_y + d_1 * v1_y)^2 = R^2 \tag{6.9}$$

$$\Rightarrow d_1{}^{\dagger} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{6.10}$$

where

$$\begin{aligned} a &= v1_x^2 + v1_y^2; \\ b &= 2(view_x v1_x + view_y v1_y); \\ c &= view_x^2 + view_y^2 - R^2. \end{aligned}$$

d_2 can be computed by performing the similar operations.

Substituting d_1 and d_2 into the corresponding equations of the rays give the intersection points, T_{start} and T_{end}.

The three control points T_{start}, T_{end}, and *viewpoint* define the visible portion of the information space and allow us to determine whether an object is visible to users.

6.1.6 Computing Hand Position

In section 6.1.5, we discussed how we determine the visible portion of the information space from the viewpoint position. In this section, we are going to discuss how the user's viewpoint and hand are coordinated, so that the user can select objects in the information space.

†If the discriminant of equation 6.9 is less than 0, no intersection point is found. If the discriminant is greater than 0, two solutions are found for d_1. The one with larger value will be selected.

To perform object selection in a 3D world, the relative position of the hand with respect to the viewpoint has to be determined. Just like the viewpoint position, discussed in section 6.1.4, the hand position consists of two parts: location and orientation. We used a tracking device to measure the hand position at different points in time. However, the measurement allows us to compute only the hand's positions relative to its initial position. It does not provide users with enough information to perform object selection in 3D space unless the relative hand position is related to the eye position. Figure 6.4 illustrates the relationship between eye and hand in object selection.

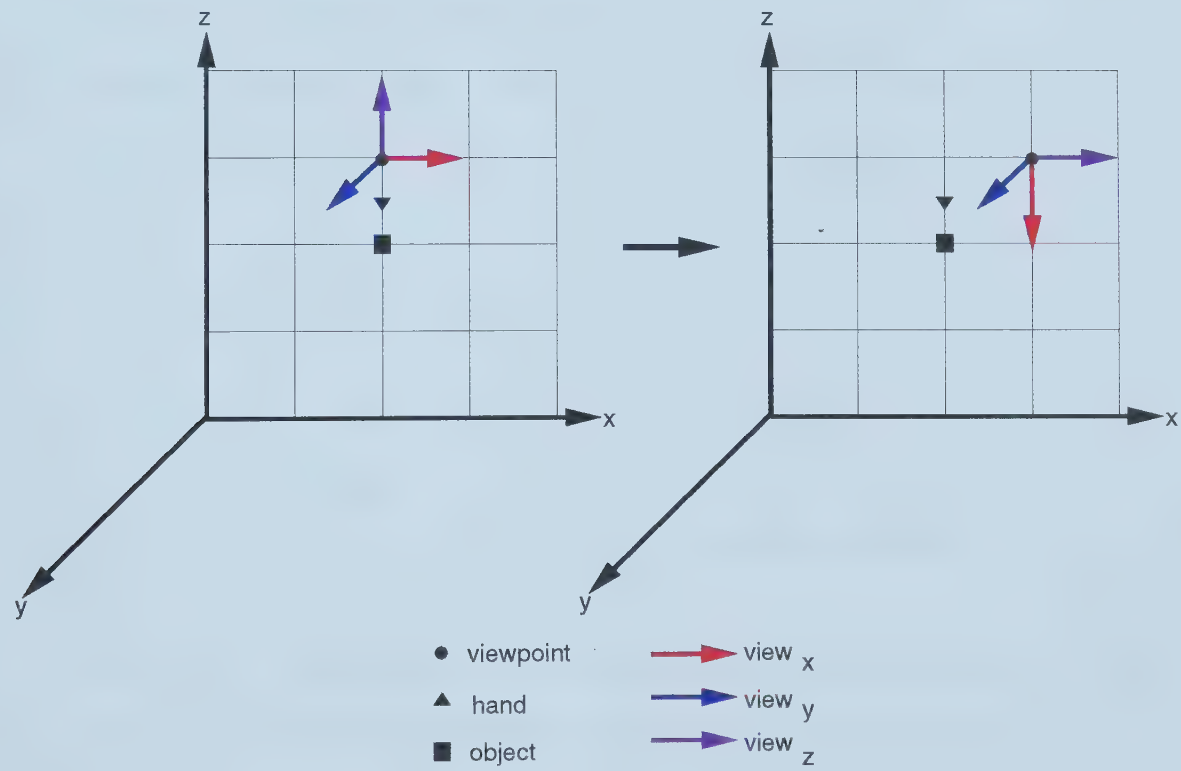


Figure 6.4: Relationship between eye and hand positions.

In figure 6.4, the *viewpoint*, *hand*, and *object* are on the xz plane. Both *hand* and *object* remain stable. They are at (2.0, 0.0, 2.5) and (2.0, 0.0, 2.0) respectively. The *viewpoint* is initially positioned at (2.0, 0.0, 3.0), and then it is moved to (3.0, 0.0, 3.0). The *viewpint* orientation is rotated by $\frac{-\pi}{2}$ about the y-axis after movement. In reality, before we perform any object selection, we estimate the object's and our hand's positions relative to our eyes and then adjust our hand position. Figure 6.4

shows that the relative positions of the object and the hand with respect to the *viewpoint* are different before and after movement, therefore, the hand needs to be adjusted differently in order to grasp the object.

In the following, we discuss how the hand's position, which includes location and orientation, is related to the *viewpoint*.

1. Hand location relative to the *viewpoint*

   The relative location of the hand with respect to the *viewpoint* depends on the viewpoint position and orientation. Figure 6.4 shows that after the *viewpoint* position has been moved, the *hand* position relative to the *viewpoint* is changed from (0.0, 0.0, -0.5) to (0.5, 0.0, -1.0).

   Suppose the hand's current location relative to its initial location is:

$$h_{location} = \begin{pmatrix} x_h \\ y_h \\ z_h \end{pmatrix};$$

   and the *viewpoint* position is at:

$$viewpoint = \begin{cases} \begin{pmatrix} x_{view} \\ y_{view} \\ z_{view} \end{pmatrix} & \text{view location;} \\ \begin{pmatrix} \omega_{view} \\ a_{view} \\ b_{veiw} \\ c_{view} \end{pmatrix} & \text{viewpoint orientation.} \end{cases}$$

   Since the *viewpoint* location is set to be an origin of the *viewpoint* coordinate system, the *hand* location must be translated by -(x_{view}, y_{view}, z_{view}).

$$\vec{vh'} = \begin{pmatrix} x_h - x_{view} \\ y_h - y_{view} \\ z_h - z_{view} \end{pmatrix}.$$

   Therefore, the hand position relative to the *viewpoint* is defined by:

$$\vec{vh'} \begin{pmatrix} \omega_{view} \\ a_{view} \\ b_{veiw} \\ c_{view} \end{pmatrix} + \begin{pmatrix} x_{view} \\ y_{view} \\ z_{view} \end{pmatrix}. \qquad (6.11)$$

2. Hand orientation relative to the *viewpoint*

   Since the *hand* position is expected to be relative to the *viewpoint*, its orientation should be changed as the *viewpoint* orientation is changed. From figure 6.4, we notice that after the *viewpoint* orientation has been rotated by $\frac{-\pi}{2}$ about the y-axis, the *hand* orientation relative to the *viewpoint* is rotated by $\frac{\pi}{2}$ about the y-axis[‡]. This means that if the *viewpoint* is rotated by ω about (a, b, c), the *hand* orientation will be rotated by -ω about the same vector.

   Suppose the hand's current orientation relative to its initial orientation is:

$$h_{orientation} = \begin{pmatrix} \omega_{hand} \\ a_{hand} \\ b_{hand} \\ c_{hand} \end{pmatrix}.$$

   The hand's orientation relative to the *viewpoint* is defined as:

$$\begin{pmatrix} \omega_{view} \\ a_{view} \\ b_{veiw} \\ c_{view} \end{pmatrix}^{-1} \begin{pmatrix} \omega_{hand} \\ a_{hand} \\ b_{hand} \\ c_{hand} \end{pmatrix}. \tag{6.12}$$

6.1.7 Semi-Transparent Rendering

Our visualization system makes use of alpha transparency to produce semi-transparent layers in a 3D graphical display. In alpha transparency, the color of each polygon is determined by using the following equation:

$$ResultingColor = \alpha * SourceColor + (1 - \alpha) * FilledColor \tag{6.13}$$

where the $SourceColor$ is the color of the new primitive, the $FilledColor$ is the color currently stored in the frame buffer, and the $ResultingColor$ is the color after modification. The α in equation 6.13 is a control variable which is used to control the transparency or opacity of the polygon. If $\alpha = 0.0$, the polygon is perfectly transparent and all polygons behind it are completely visible. If $\alpha = 1.0$, the polygon is opaque and completely occludes polygons behind it. As the value of α ranges from 0.0 to 1.0, the value of $ResultingColor$ varies smoothly from $FilledColor$ to

[‡]Assume both *hand* and *viewpoint* have the same orientation at the beginning.

$SourceColor$, and the transparency of the polygon decreases gradually. By changing the value of α, we can adjust the opacity of the object.

Different values of α will produce different degrees of transparency. The question is how to determine the value of α. The alpha transparency procedure in the HP Starbase graphics library [32] provides us with two modes for determining the value of α. If we use the Starbase command `alpha_transparency(fildes, mode,` ρ `, min, max)`, the second parameter `mode` is used to specify how the value of α is determined. If `mode` = 1, α will be equal to the value of `min` where `min` is the lowest transparent degree we want. This method is very simple but not realistic. It assigns an entire polygon the same degree of transparency no matter what direction the user is looking from.

In order to have a more realistic display, the value of α is determined by using a more sophisticated method in which the direction to the viewpoint is considered. If `mode` in `alpha_transparency` equals to 2, the value of α will be computed using the following equation:

$$\alpha = max - (max - min) * \cos^{\rho} \theta \tag{6.14}$$

where max and min are the highest and lowest transparent levels we want, θ denotes the angle between the surface normal and the direction to the viewpoint, and ρ specifies how much the $\cos\theta$ affects the value of α. Both max and min must be within the range of 0.0 and 1.0, and min is smaller or equal to max. ρ is any integer which is between 1 and 16,384. Since θ is between -$\frac{pi}{2}$ and $\frac{pi}{2}$, the second term in the right hand size of equation 6.14 must be between 0.0 and (max - min). Thus, α is between min and max. As $|\theta|$ increases, the value of α is increased. This implies that as the angle between the surface normal and the direction to the viewpoint increases, the transparent degree of the object is decreased. And users perceive different degrees of transparency when they look from different directions.

In order to have proper results in images with transparent surfaces, the transparent layers must be rendered in the correct order. Whenever a polygon is rendered, the Z-buffer values must be updated. If the transparent polygons are rendered from front to back with respect to the eyepoint, the Z-buffer values for the foreground polygon will be updated first. The α value of the background polygon will affect the foreground

polygon. Therefore, the order of rendering is very important.

In our case, however, rings are one above the other, but never intersect each other. For this reason, we can determine the order of rendering based on how far the layer is away from the center of the ring. If rings are drawn in the right order, the Z-buffer also works well for translucent objects as well as for solid objects.

6.2 System Model

In order to understand how the system generates a graphical view of the data, we need to have a good understanding of our system design. Figure 6.5 shows the basic structure of our system. In this section, we describe the main components of our system and how they interact to present and update a graphical view of the data.

6.2.1 System Components

In order to make the program easy to extend[§], an object-oriented approach is used in its implementation. Each rectangular box in figure 6.5 represents an object in our system. Based on the set of operations performed by the objects, the system is seperated into two components, bounded by dotted lines in figure 6.5.

6.2.1.1 Component One

The objects in this component are mainly responsible for controlling the interaction with users. They handle the interaction of the system with users and the interaction of objects with each other. Two objects in this component are Panels and Main.

6.2.1.2 Component Two

The objects in this component are mainly responsible for the computation, geometric modelling, and presentation of the visualization. They manage all non-graphical computations in the application, converting the computations into a graphical representation, and finally present it on the output device. The objects in this component are described as follows.

[§] Figure 6.5 shows that a new data set can be added by simply creating a new object in Group 1. Similarly, a new 3D object can be included in the display by adding a new object in Group 2.

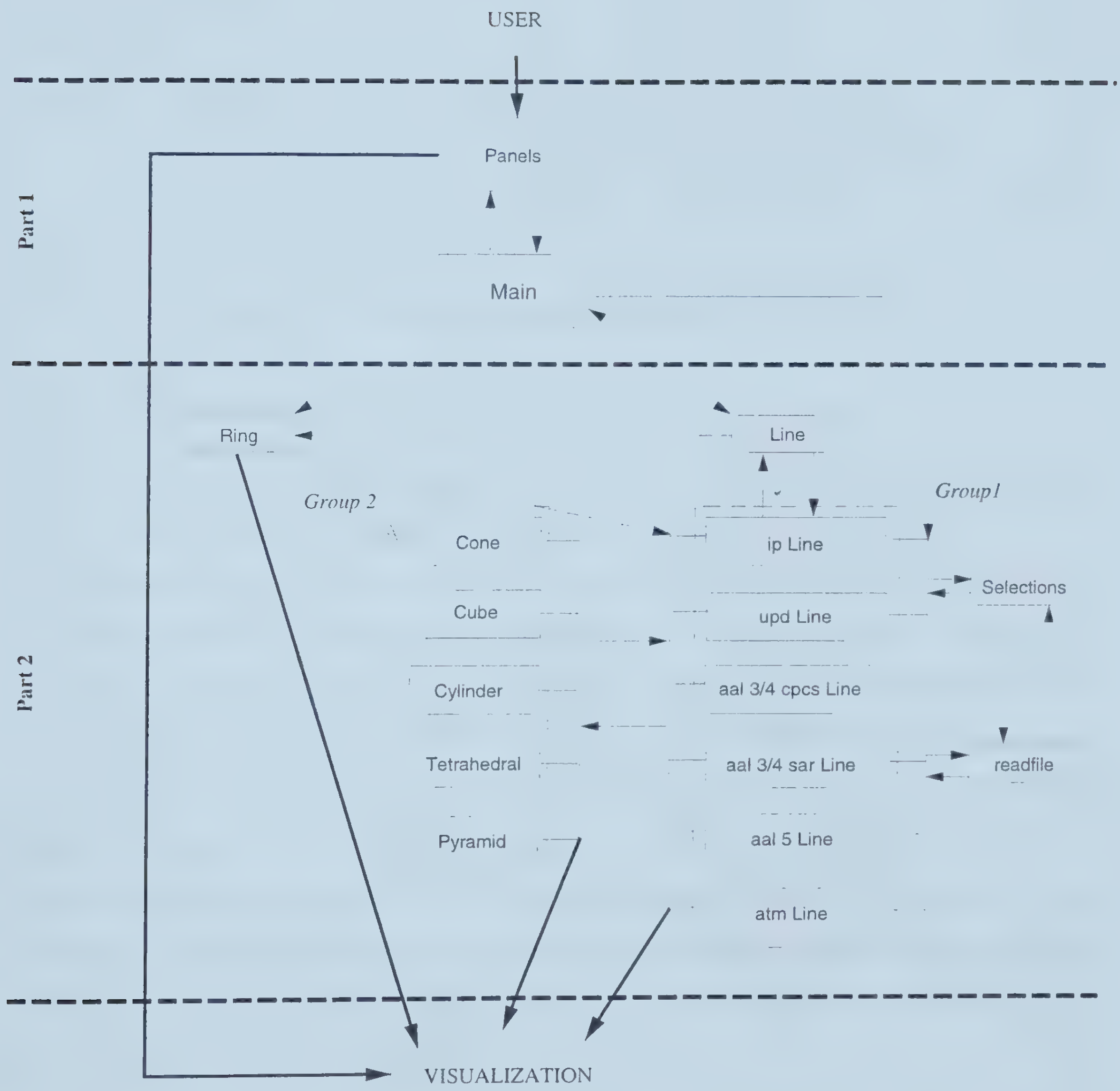


Figure 6.5: Our System Model.

Objects	Functions
Panels	• constructs and draws all required panels on the output window. • accepts requests from the user. • gives response to the user, and transmits the requests to the main control.
Main	• constructs a 3D virtual environment for visualizations. • interacts with other objects to manage the user's requests. • updates the display including recomputing the viewpoint positon and the visible portion.

Table 6.1: Objects in the first component of the system.

6.2.2 Interaction between Objects

After we have described each object in our system, we now explain what happens if a user makes a request. Assume the request is to visualize the AAL 3/4 SAR data set. The event flow between different objects is represented by the arrows in figure 6.5. The italic words in the following paragraph indicate the objects in figure 6.5.

After the user has started up the system, the *Panels* accept the request and sends it to the *Main*, which forwards this request to the *Line*. This object will invoke the *aal 3/4 sar Line* to create a line of AAL 3/4 SAR data. The *aal 3/4 sar Line* will construct a line for AAL 3/4 SAR data by reading the data from the *readfile* and determining the color and vertical positon from the *Selections*. After the AAL 3/4 SAR line has been created, it is sent back from *aal 3/4 sar Line* to *Line*, and finally to *Main*. In the next display update cycle, the *Main* will invoke *aal 3/4 sar Line* to draw the line in the output window.

6.3 The Prototype System

Our prototype system can support visualization of ATM protocol data with any number of data and protocol layers. The ring and the graphical objects are displayed at the bottom part of the window, The top part is for user interface.

The user interface for our system consists of three panels. They are :

Objects	Functions
Ring	• defines a ring including its radius, and size. • labels the ring. • determines and draws the portion of the ring which will be included in the display.
Line	• obtains a "line" from atm, aal 3/4 cpcs, aal 3/4 sar, aal5, ip, and udp lines. In our implementation, each "line" consists of a set of cells. The cells represent the data in the data file. Therefore, each line should have a large number of cells. Each cell stores all important information in each data item.
atm, aal 3/4 CPCS, aal 3/4 SAR, aal 5, ip, and udp	• depends on the the data type to construct its graphical representation. • searches for a particular data in a particular line. • draws a 2D graphical representation of the data.
Cone, Cube, Cylinder, Tetrahedral, and Pyramid	• constructs and draws a 3D object.
Selections	• determines the color, and vertical position of a particular data.
Readfile	• reads in the data, such as ATM cells, AAL protocol data, and higher level protocol data, from a data file. • stores all useful information. • calculates and stores the vertical location of each data item.

Table 6.2: Objects in the second component of the system.

Main Panel :

It contains three buttons and three pull-down menus, as shown in Plate 6.2. They are used to invoke different operations in the system. Table 6.3 describe the operations invoked by each button and each menu.

Name	Type	Operation
Quit	button	• To terminate the visualization process.
Compressed (On/Off)	button	• If this button is down or the button is in "On" state, the portion with no data is compressed.
Level (Upper/Lower)	button	• The ring is seperated into two levels, upper and lower. A sequence of graphical objects is lying in either the upper or lower level of the ring. The main reason for having two levels is that two graphical objects can be placed seperatedly, instead of one overlapping the other, if they are very similar in shape.
Layer Selection	pull-down menu	• To provide a different type of data. User can select which type of data they want to visualize.
Transparency	pull-down menu	• Users can adjust the transparency of a particular layer.
PDU	pull-down menu	• To provide a list of data type included in the visualization. Users can select the type that they want to get a detailed description of.

Table 6.3: Six operations in the Main panel.

Transparency Panel :

It has slider that allows users to control the degree of transparency in Equation

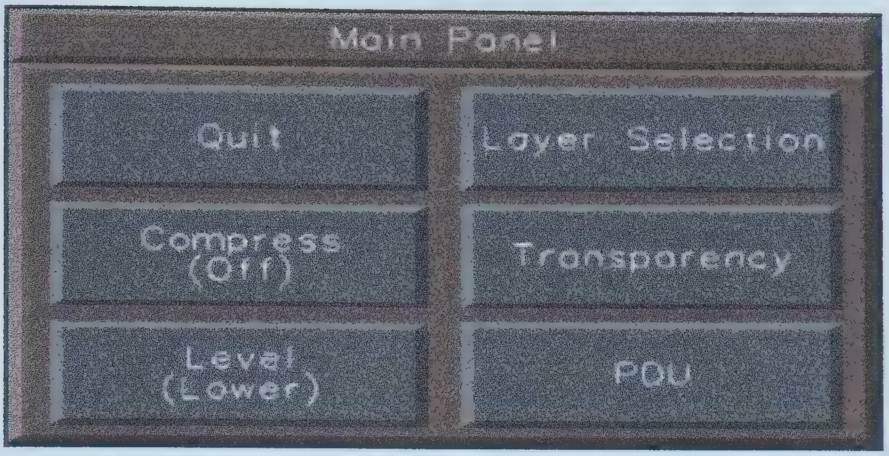


Plate 6.2: Main Panel.

6.14. See Figure 6.3

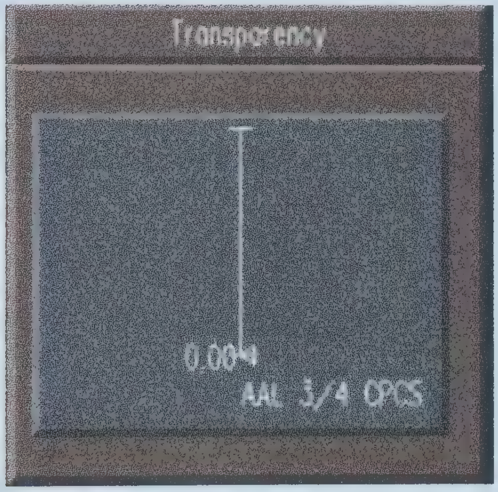
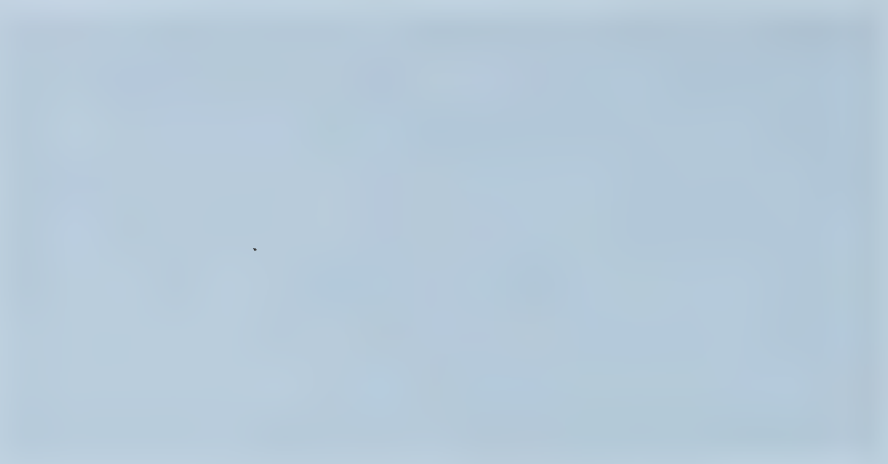
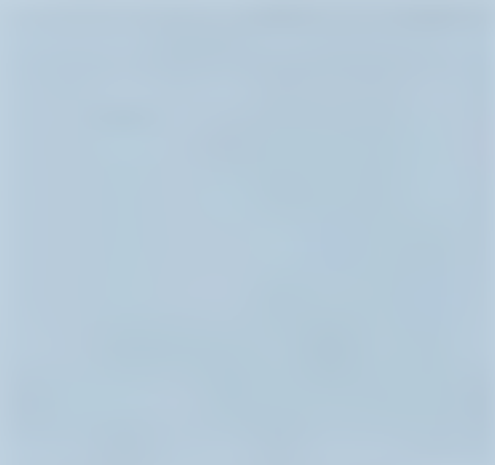


Plate 6.3: Transparency Panel.

PDU Panel :

It contains a text area along with two buttons. The two buttons, forward and backward, are used to select the cell which is previous to or next to the currently selected cell. The text area is used to show the details of a particular cell in a particular layer. See Figure 6.4.





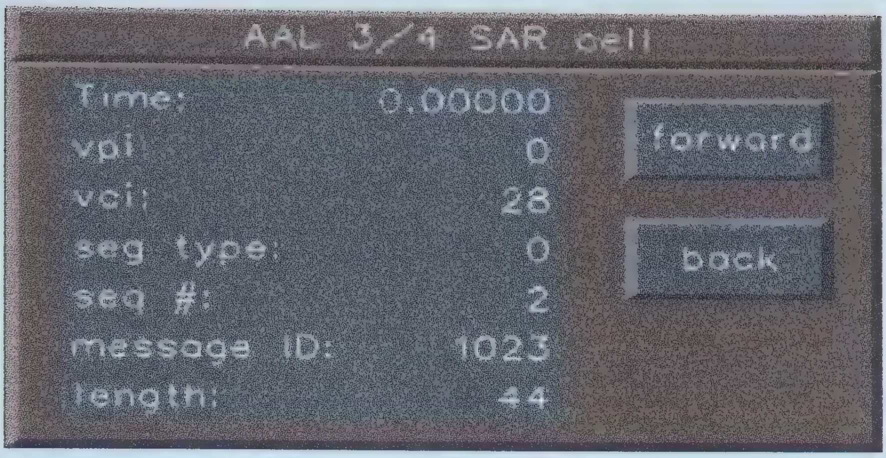


Plate 6.4: PDU Panel.

6.3.1 Response Time

The prototype system is able to maintain real time response on an HP 9000 workstation for a display with more than 1000 pieces of data. The table below summarizes the update rates of our system with different numbers of data involved in the visualizations.

Data Type	Sum of Data	Sum of Polygons	Update rate ($\frac{\#frames}{sec}$) with	
			normal ring	compressed ring
atm cells +	100	200	16.054	11.355
aal 3/4 sar +	200	400	12.087	11.270
aal 3/4 cpcs +	208	408	11.521	11.219
ip +	308	708	11.419	11.184
udp +	338	768	10.955	10.216

Table 6.4: Update rates with different numbers of data.

In Table 6.4, the first column is the data type involved in the visualization. The second and third columns are the total numbers of data and polygons drawn. The last two columns are the update rates with normal and compressed rings respectively.

6.3.2 System Notes

The prototype system is implemented in C++ and on top of the Minimal Reality (MR) Toolkit [11, 25] with the HP Starbase Graphics Library [32]. The MR Toolkit is a toolkit for Virtual Reality (VR) applications developed at the University of Alberta.

1. Display Device :

   Our system only supports a conventional CRT. The entire information structure and the three panels are projected onto a 780x1024 2D display area.

2. Input Device :

   Two six degrees freedom trackers [23] are used as input devices for our prototype system. One controls the location and orientation of the viewpoint, and the other is used for interacting with the 3D panels.

   There are a lot of input devices available such as a keyboard, mouse, or joystick. However, none of them support direct object manipulation and navigation in 3D space. Figure 6.5 shows the problems caused by using these three input devices in 3D space.

   Since direct interaction and navigation are very important in our system, the three input devices in Figure 6.5 are not suitable for our system. With the use of the tracker, the user's position can be detected by the sensor which is attached to the user. No manual input is required. In addition, trackers can provide up to six degrees of freedom. Therefore, the user does not need to break down a 3D task into several lower dimensional subtasks. More importantly, movements can be invoked by simply moving the tracker's position and clicking on the tracker buttons. How fast the tracker moves has no effect on the cursor movement, and the action of clicking is a common action used in all 2D direct manipulation systems. Therefore, motion can be easily controled and performed. These are the reasons we chose 3D trackers as input devices for our prototype system.

Name	Type	Input Style	Problem
keyboard	zero-dimensions	The user types in the values of location and orientation via keyboard.	• typing is cumbersome, error prone and slow.
Mouse	two-dimensions	The user specifies the the location and orientation by performing pointing and clicking actions.	• It can not support a single translation along all three directions. So translation along three directions must be broken down into sub-tasks. Besides, it limits the rotations to around one of the major axes at a given time. Due to the fact that rotations are not commutative, the order of rotations plays an important role.
Joystick	three-dimensions	The user controls the cursor's position by moving the stick forward, backward, and around.	• Since the cursor speed depends on how much force is applied, motion control is very difficult.

Table 6.5: Comparsion of three different input devices.

Chapter 7

Evaluations

Previously, we discussed the problems with visualizing ATM protocol data and presented some possible approaches to these problems. In this chapter, we compare our 3D visualization system with the 2D system. In doing so, we highlight some of the major differences, advantages and disadvantages of our 3D system compared to the 2D one.

7.1 Presentation of a Single Data Set

Differences

The 2D and 3D approaches make use of different mapping mechanisms to solve the problem of single data presentation. Basically, both 2D and 3D visualization systems use the same approach, including determining what information is interesting and useful, considering which graphical properties are available, and mapping the properties to the selected information. The main difference between 2D and 3D systems is that they have different mapping mechanisms. In the 2D approach, two sets of coding, colour and shape, are used at the same time to emphasize the difference between PDUs at different protocol layers. The inside shape of each graphical object is used to encode the payload part of ATM cells or AAL PDUs. In the 3D approach, different shapes are used to encode different types of data attributes, and the color of each shape represents the value of the attribute. Therefore, two objects having the same shape with the same color implies that they represent the same attribute with the same value.

Advantages

The 3D approach has an advantage: the mapping mechanism used in this approach allows more information to be encoded in a single graphical unit. Different shapes are combined to form one graphical unit, and each shape represents one data attribute. If a graphical unit consists of three different shapes, there will be three types of information encoded in the unit. However, in the 2D approach, only two types of information are provided in each graphical unit.

Disadvantages

The main disadvantage of this 3D approach is that using colour to encode the value of a data attribute may not be appropriate. Although there are many different colours available, humans are limited in their ability to differentiate colours. Some data attributes may have many possible data values. Using colours to encode these values results in displays with a lot of distractions, thereby lowering the efficiency of the visual effect in the presentation.

7.2 Presentation of Large Data Set

Differences

The 2D visualization system uses three time lines to visualize the protocol data at three different scales. The top most time line is used to present the entire content of the capture buffer at a low level of detail. Each unit in this time line represents several hundred ATM cells, and the colouring of the unit shows the density of cells over the time interval. The next time line shows a more detailed view of part of the capture buffer. At this level, aggregate information is still displayed, but the locations of error cells are indicated by red lines. The bottom time line is a detailed view of the cells in the capture buffer. At this level, a separate graphical representation is produced for each cell. This representation covers the interval of time the cell occupies, and its colouring or geometry shows the cell's or PDU's important properties. Due to the size of the capture buffer, the data stored in the buffer cannot be displayed within a single display at the same time. The system provides the user with scrolling. With

this technique, the user can scroll through the time line in order to have a complete view of the buffer.

The approach taken by our visualization system is completely different from the 2D approach because our system presents data in 3D space. A particular problem in the visualization of ATM protocol data is inadequate display space. Moving to 3D space provides a promising solution to this problem. The idea is to lay out the entire content of the capture buffer on a ring's surface. The level of detail shown depends on the distance between the ring and the viewpoint. The closer to the ring, the higher the level of detail a viewer can perceive. The viewer can get a view of the entire buffer by doing a complete rotation about the center of the ring.

Advantages

This new approach has several advantages over the 2D one.

1. Focusing on a particular cell while keeping scene context

   Scrolling is widely used in the 2D system. By scrolling the top most time line, users can visualize the complete capture buffer. Scrolling the top two time lines alternatively allows users to find areas of interest in the buffer. To examine a particular cell in the buffer, user can scroll through the bottom time line. Undeniably, scrolling is a very useful mechanism. With scrolling, no matter how large the capture buffer is, the complete buffer content can still be included in a limited 2D display. However, in order to visualize the complete content of the capture buffer and to focus on specific regions concurrently, users must scroll through three different time lines individually. This results in lowering the efficiency of the system.

   In our 3D approach, the entire contents of the capture buffer are mapped around the surface of the ring and can be seen by rotating about the center of the ring. To get a detailed view of a particular cell in the buffer, users first select the appropriate protocol layer from the main panel, and then click on the cell. A small panel, along with the description of the selected cell, will be popped up above the ring. This approach not only integrates focus and context in a single

view but also provides users with an efficient tool to control areas of interest in a huge information space.

2. Continuous level of detail

   In the 2D approach, there are three different levels of detail. The top most time line presents a high level representation of the entire buffer content. The next level time line provides more detailed information of each data package. The bottom time line is a graphical representation of each data. These three levels of detail are neither continuous nor smooth. Users need to mentally construct relationships among three levels.

   In our 3D approach, there are not only multiple levels of detail, but also continuous changes across different levels of detail. Our 3D visualization simulates the natural phenomenon in the real world. In the real world, an object gradually become bigger and bigger as we come closer to it. We have a more detailed view of the object as we get closer to it. In our visualization environment, the distance between the viewpoint and the object controls the level of detail that the user perceives, and changes between different levels of detail are smooth and continuous. The smooth transitions among different levels of detail provided by the 3D approach prevent abrupt changes in the object and help users to maintain context when displaying views.

3. Ease and flexibility in reaching a cell

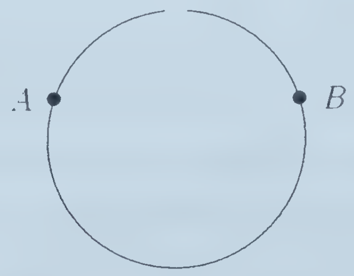


Figure 7.1: 3D Approach

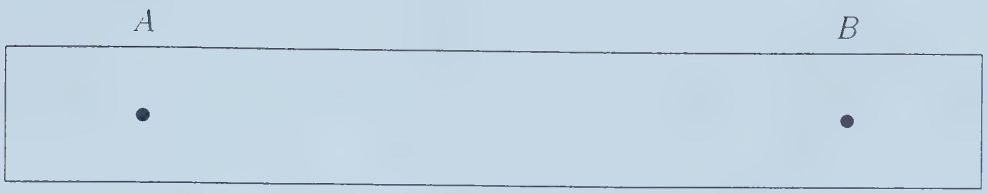


Figure 7.2: 2D Approach

In the 2D approach, there is only one way to go from one point to another, whereas two possible paths are available for each move in the 3D approach. To give a precise explanation of this idea, let us look at Figures 7.1 and 7.2, which illustrate data searchings in the 3D and 2D systems.

Assume A is the current position and B is the final position. To move from point A to point B, we must go right in the 2D approach; however, we can rotate either clockwise or anticlockwise in the 3D approach. This implies that the 3D approach is more flexible than the 2D one.

Assume points A and B in Figure 7.1 are the same as the ones in Figure 7.2. The minimum distance between A and B in the 3D approach is smaller than the distance in the 2D approach. If the display screens in both approaches are the same size, the chance that both points A and B are included in the same view will be higher in the 3D approach than in the 2D approach. This means that it is easier to visualize points A and B at once in the 3D approach than in the 2D one.

4. No scrolling is needed

   In the 2D approach, multiple layers of the protocol stack are stacked one above the others. Users may have to scroll vertically in order to visualize a particular layer. In the case of examining the relationship among different data units at different protocol layers, users have to scroll vertically and horizontally.

   In our 3D approach, multiple layers of the protocol stack are stacked one on top of the other. With the use of transparency, multiple layers of protocol layers can be visible at the same time without any scrolling. Users can easily examine how a data unit in one layer is related to other layers.

5. Better information arrangement

   Another point we observe in the previous example is that for the same amount of information the 3D approach requires a smaller display space than the 2D approach does. Therefore, a better arrangement of information can be provided by using the 3D approach.

Advantage of Using Ring Structure

One of the main advantages of the ring structure is that users have control over the level of detail. In the 2D approach [18], users were presented with three time lines that gave three different levels of detail; however, users had no control over how much detail was shown in each time line. In the case of the ring, the user can interactively adjust the level of detail. As the user moves far away from the ring's surface, he or she sees a high level representation of the capture buffer, similar to that seen in the first time line of the two dimensional visualizations. On the other hand, when the user moves closer to the ring's surface, more details of that portion of the ring become visible. As long as the user is close enough to the surface, the details of the individual cells are visible. With this layout, the user can control the amount of detail by simply moving backwards and forwards in space. This variable level of detail is one of the main benefits of the ring structure.

Disadvantages

The main disadvantage of this 3D approach is that the flat display, especially at a static stage, sometimes misleads the user. When the viewpoint keeps changing, there are more depth cues presented, including occlusion and relative size cues. However, when the viewpoint is stable, the depth cues in the display are not so obvious to the user. Due to the lack of depth cues, the display looks flat. The flat display makes it more difficult for the user to realize the information is displayed in 3D space.

7.3 Presentation of Multiple Protocol Layers

Differences

The 2D approach solves the problem of presenting multiple protocol layers by placing all protocol layers in the bottom time line. The visualization still consists of three time lines. The top two time lines do not change, but the bottom time line may have more than one layer of information displayed. If there is not enough space for all protocol layers, scrolling will be applied to solve the space limitation problem. By scrolling through multiple layers of protocol data, the user can visualize a certain number of protocol layers at once.

Our 3D approach maximizes the use of limited display space by placing layers of information one on top of another or one above another. Layers of information are organized as ring structures in a 3D space to facilitate information arrangement, as in Chapter 5. In this way, more information can be included in a single display screen. However, not all the information included is visible because the foreground objects occlude the background objects. Our 3D approach supports the use of transparency to expose all background objects. Therefore, all information in the display screen is visible to the user.

Advantages

This approach has several advantages over the 2D approach.

1. Multiple levels of detail for all protocol layers

   In the 2D approach, only the ATM cell layer has three different levels of detail, and all the others have a single level of detail. Our 3D approach provides all protocol layers included in the visualization with multiple levels of detail. The user simply adjusts the distance between the viewpoint and the ring's surface to change the level of detail for each protocol layer.

2. Enough space for an arbitrary number of protocol layers

   In the 2D approach, the number of protocol layers that can be displayed in a single view is restricted by the amount of space available for the bottom

time line. Scrolling allows more protocol layers to be included in the entire visualization, but not in a single view. For a deeply nested protocol stack, the user may need to scroll through several levels of the protocol layers. In the case of searching for a particular unit in a particular protocol layer, the user may need to scroll through the bottom time line and the multiple protocol layers. All this scrolling complicates the visualization task and lowers the efficiency of the system. The 3D approach makes use of the stacking technique to arrange multiple protocol layers in a single display; therefore, no matter how deep the protocol stack is nested, all protocol layers can be included and visualized at once without any scrolling.

Disadvantages

Users may choose to stack all protocol layers on top of each other only in the vertical direction or the horizontal direction. If the number of protocol layers involved is not too large, both choices are acceptable. Otherwise, problems may occur.

1. Inefficient

   Stacking all protocol layers on top of each other in the vertical direction only is not efficient. Since this type of stacking is one dimensional, the number of protocol layers that can be stacked is still restricted by the limit of the vertical height. Even if there is no limit to the number of layers shown in this way, tracing data through multiple protocol layers is very time consuming. Therefore, this type of stacking is not so good.

2. Misleading users

   Stacking all protocol layers on top of each other in the horizontal direction may confuse users. With this type of stacking, the resulting display will have a number of concentric rings with the inner ring having a smaller radius than the outer ring, and thus the objects on the inner ring will be rendered larger than objects on the outer ring. If the display consists of a large number of such concentric rings, the distance between the inner and the outer rings will be very large. This implies that even if the objects on these two different rings are of

the same size, the objects on the inner ring will still be rendered larger than the objects on the outer ring. This result could mislead users about the sizes of the objects.

7.4 Interactions

Differences

Three basic operations are provided in both visualization systems. However, these systems use different techniques to implement them.

1. Protocol Layer Selection

   The actions of clicking and dragging are used in 2D and 3D systems to select the different protocol layers involved in a visualization. The main differences are that they are performed in different spaces with different input devices.

2. Area of Interest Searching

   The 2D approach makes use of a scrolling mechanism to locate an area of interest, whereas the 3D approach supports interactive rotation for navigating around the 3D space.

3. Description of each data unit

   Both systems provide users with complete descriptions of cells present in the visualizations. In the 2D system, a complete record of the entire buffer content is shown on a 2D popup window. In the 3D system, a description of a particular cell is displayed on a 3D panel.

Other than these three basic operations, our 3D system supports many other interactions such as target acquisition, target designation, degree of transparency, and level of detail control.

Advantages

The interactions provided by the 3D system provide users with greater insights into the protocol data. Being able to navigate around the information space, the user

can see an overview of the data and the cell distribution over time. Zooming into a particular region enables users to focus on a smaller region. Coupling these with other user interface controls such as level of detail, area of interest, and translucency can further enhance the user's perception of the data.

Disadvantages

The main problem with the interaction techniques provided by the 3D system is that users require some practice in order to gain efficiency in using them. The main reason for this is that sometimes the target is not easy to reach for the following reasons.

1. the setup of the input device

   One of the factors affecting the sensitivity of the input device is the relative positions of the sensors and source. In order to have precise detection, the sensors and the source must be correctly positioned. Therefore, after the sensor has been set up, the user must find a location where both source and sensors can achieve optimal performance.

2. human characteristics

   Some people cannot keep holding an object with one hand for a long time without any hand shaking motion. This characteristic does not have as great an influence when a mouse or keyboard is used, since the hands are placed on the table. However, this is not the case in our visualization system. The input devices we use are a pair of trackers, one controlling the position and orientation of the viewpoint, and the other making selections from the panels. The user has a choice to not place his hand on the table. In this case, the target acquisition will be greatly affected if the user's hands keep shaking.

Chapter 8

Future Works

In this chapter, we outline the work currently in progress and some ideas for future research :

1. Improving the current system

   We are working to improve the quality of the existing 3D approaches. Although our 3D system has allowed us to detect and determine the source of errors in a stream of PDUs in ways not easily possible via other mechanisms, such as the 2D system and the text-based system, some issues still need to be improved.

   - visual representation

     Our system uses color to encode the value of data attributes. However, if the number of different data values increases, the number of colors demanded increases. At present, the system assigns an arbitrary color to a data value. The ease of Differentiating among colors may not be as easy as we thought, especially as the number of colors involved is large. We plan to investigate some color assignment allocation schemes which can reduce the confusion caused by some inappropriate color assignments.

   - information identification

     Our system can process and display multiple layers of protocol stack data. However, as the number of protocol stack layers increases, the number of graphical objects stacking over each other is increased. Identifying different types of objects from the display becomes very difficult. We plan to offer a "help" option in the display window which provides users with some

description about different graphical objects. In addition, we plan to use luminosity on the layer as the user moves through different layers.

- Navigation

  Another limitation of our 3D system is the inability to make any marker on the ring. In our visualization, times serve as addresses in a virtual world. Users simply remember the time in order to return back to the same position on the ring. If there are a few possible positions that the user may want to return to, this is definitely not a good approach. We plan to allow the user to make landmarks on the ring so that he can return to any marked position.

- Flexibility

  When constructing a scene, it is not possible to undo a step. For example, after the user has placed a layer on top of another layer, he cannot change his decision. By being able to undo a step, he can re-consider where to put the layer.

2. Usability Studies

   Another limitation of our system is that no formal evaluation of how useful our views really are have been done so far. We intend to perform user studies in the near future. These studies will not only can evaluate the effectiveness of our visualization techniques but also give us new insights into what will help to improve our system.

3. Additional Visualization Tasks

   One future activity is determining how the significant advantages of our 3D approaches reported here will generalize across different levels of task complexity.

   - Stream Comparison Visualizations

     A number of tests involve taking a stream of cells, feeding them through a device, and collecting the device's output as a second stream of cells. The user then wants to determine whether the device has properly processed the input stream. At the present time there is no easy way of making this

comparison, even though the input and output streams can both be captured. We need to develop visualization techniques that allow the user to compare two streams of cells to determine whether the device is functioning correctly. These visualization techniques should highlight the timing relationships between the two streams, dropped cells, inserted cells, cells with address and other header errors, and errors in the payload. With the ring structure in our 3D system, two streams of cells can be stacked one on top of or above each other. By rotating around the ring, users can compare between two streams of cells.

- Network Management Visualizations

  Our current visualization techniques mainly address the problems of detecting errors and determining the source of errors in protocol stack layers. We plan to investigate how the techniques can be modified so that they can be used in network management visualization.

4. Increasing Availability

   At present, our prototype is restricted to running on HP workstations, using Starbase as the graphics API and two 3D trackers as interaction devices. The new prototype will run on PC workstations, using Window NT as the main platform, and using OpenGL as the graphics API. In order to allow the new prototype to run on different platforms, MRObjects [12] is used. MRObjects is an object oriented framework for the development of VR applications. It has its own high level graphics API and a set of VR interaction techniques. With MRObjects, the new prototype will be able to run on PC workstations, using OpenGL as the graphics API and having a wider range of input devices.

Chapter 9

Conclusions

The thesis began with an introduction to the limitations of the time line visualization in the 2D system [18] and the goals to be achieved by this thesis. The time line visualization makes use of hierarchies to display data at several levels of detail and scrolling to view the complete capture buffer. It provides users with an efficient way for visualizing large volumes of data, and is very useful in detecting errors and performance problems in the data collected by the HP Boardband Series Testing System. This technqiue works well as only one level of the protocol stack is examined. However, as multiple levels of the protocol stack are involved in visualization, maintaining multiple levels of detail in each protocol stack layer results in a space limitation problem. The main goals of this thesis is to design visualization techniques which solve the limitation in the 2D system.

Chapters 2 and 3 gave overviews of previous work in the area of information visualization and some background on ATM, including ATM technology, the ATM protocol reference model, and ATM networks.

Chapters 4 through 7 described my original work. In Chapter 4, we described the problems associated with ATM protocol data visualization. Generally, there are two problems involved in designing a visualization system for ATM protocol data: inventing an adequate representation of the data and helpful interaction techniques. In Chapter 5, based on the problems identified in Chapter 4, we designed a set of visualization and interaction techniques. The chief difficulty for visualizing ATM protocol data is that the underlying information structure is large and complex. One of the best ways to understand this type of information structure is to provide users

with a greater context. To achieve this, we optimized the use of display by placing information around the inside surface of a ring in 3D space. In the case of multiple protocol layers, the layers are stacked one on top of the other in the vertical or horizontal direction. Through the use of transparency, all objects except the ones on the outer most ring become partially transparent so that background objects are visible through foreground objects. In this way, no matter how many protocol layers there are and how complex the underlying structure is, there is still enough space to visualize all information in a single display at the same time. The use of transparency also allows users to arbitrarily adjust the significance of the information. In addition, changing the distance between the viewpoint and the ring's surface produces different perspectives in 3D space to provide information with continuous and multiple levels of detail. In Chapter 6, we presented the implementation of our approaches. In Chapter 7, we made comparisons between our 3D system and the 2D one. The main problem with using 2D system to visualize ATM protocol data is inadequate display space. Moving from 2D to 3D space allows more information to be visible at once. Through the use of transparency, the 3D system further maximizes the efficient use of display space. The increased amount of information provided by the 3D system facilitates the users' understanding of a large information space, thereby facilitating error detection and performance analysis of ATM networks.

The thesis concludes with an outline of work currently in progress and some ideas for future research in Chapter 8.

Bibliography

[1] ATM Forum,
http://www.atmforum.com.

[2] Asynchronous Transfer Mode,
http://www.wavc.org/UNIVERCD/data/doc/cintrnet/ito/55755.html

[3] Christopher Ahlberg, and Ben Shneiderman. Visual Information Seeking: Tight coupling of dynamic query filter with starfield displays. In *Proceedings of ACM CHI '94 Conference*, (April 1994), pages 313-317 + color plates.

[4] Christopher Ahlberg, and Ben Shneiderman. Alphaslider: A compact and rapid selector. In *Proceedings of ACM CHI '94 Conference*, (April 1994), pages 365-371.

[5] Benjamin Bederson, and James Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proceedings of UIST '94*, pages 17-26, 1994.

[6] Eric Bier, Maureen Stone, Ken Pier, William Buxton, and Tony DeRose. Toolglass and Magic Lenses: The See-through Interface. In *Proceedings of SIGGRAPH '93, Computer Graphics Annual Conference Series*, pages 73-80.

[7] Stuart Card, George Robertson, and Jock Mackinlay. The Information Visualizer, An Information Workspace. In *CHI '91 Conference on Human Factors in Computing Systems*, pages 181-194.

[8] Jeromy Carriere, and Rick Kazman. Interacting with Huge Hierarchies: Beyond Cone Trees. In *Proceedings of Information Visualization '95 Symposium*, (October 30-31), pages 74-81.

[9] Kenneth Cox, and Stephen Eick. 3D Displays of Internet Traffic. In *Proceedings of Information Visualization '95 Symposium*, (October 30-31), pages 129-131.

[10] George W. Furnas. Generalized fisheye view. In *CHI '86 Conference on Human Factors in Computing Systems*, pages 18-23, 1986.

[11] Mark Green, and Dani Beaubien. Minimal Reality Toolkit Version 1.2 Programmer's Manual. Department of Computing Science, University of Alberta, Edmonton, Alberta, 1992.

[12] Mark Green. MRObjects http://www.cs.ualberta.ca/ graphics/mrobjects/

[13] Mark Green, and Chris Shaw. The DataPaper: Living in the Virtual World. In *Proceedings of Graphics Interface* pages 123-130, May 1990.

[14] Ninad Jog, and Ben Shneiderman. Starfield Information Visualization with Interactive Smooth Zooming. In *University of Maryland Technical Reports*, CS-TR-3286, 1994.

[15] Brian Johnson, and Ben Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings IEEE Visualization 91'* (October 1991), pages 284-291.

[16] John Lamping, Ramana Rao, and Peter Pirolli. A focus+content technique based on hyperbolic geometry for viewing large hierarchies. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, (May 1995).

[17] Jiandong Liang, and Mark Green. Geometric Modeling Using Six Degrees of Freedom Input Devices. ftp://ftp.cs.ualberta.ca/pub/graphics/papers/jdcad.ps.gz

[18] Luo Feng. Visualization of ATM Traffic. 1994 Master Thesis in University of Alberta.

[19] Jock Mackinlay, George Robertson, and Stuart Card. The Perspective Wall: Detail and context smoothly integrated. In *CHI '91 Conference on Human Factors in Computing Systems*, pages 173-179.

[20] Vibhu Mittal, Steven Roth, Johanna Moore, Joe Mattis, and Giuseppe Carenini. Generating Explanatory Captions for Information Graphics. In *Proceedings International Joint Conference on Artificial Intelligencei*, (August 1995), pages 1276-1283.

[21] Tamara Munzner. Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space. In *Proceedings of VRML '95, (San Diego, California, December 14-15, 1995), special issue of Computer Graphics, ACM SIGGRAPH*, pages 33-38.

[22] Liao Osada, and Ben Shneiderman. AlphaSlider: Searching textual lists with sliders. In *Proceedings of the Ninth Annual Japanese Conference on Human Interface*, (October 1993).

[23] Polhemus, Inc., Colchester, Vermont. *3SPACE ISOTRAK User's Manual*, 1987.

[24] Manojit Sarkar, and Marc Brown. Graphical Fisheye Views of Graphs. In *Proceedings of the ACM SIGCHI '92 Conference on Human Factors in Computer Systems*, (May 1992), pages 83-91.

[25] Chris Shaw, Jiandong Liang, Mark Green, and Yunqi Sun. The Decoupled Simulation Model for Virtual Reality Systems. In *CHI '92 Conference Proceedings*, 1992.

[26] Jun Rekimoto, and Mark Green. The Information Cube: Using Transparency in 3D Information visualization. In *In Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, pages 125-132.

[27] George Robertson, Jock Mackinlay, and Stusrt Card. Cone Trees: Animated 3D visualizations of hierarchical information. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, (April 1991), pages 189-194.

[28] Manojit Sarkar, Scott Snibbe, Oren Tversky, and Steven Reiss. Streching the Rubber Sheet: A metaphor for viewing large layouts on small screens. In *Proceedings of UISR '93*, pages 81-91.

[29] Manojit Sarkar, and Marc Brown. Graphical fisheye view. In *Communications of the ACM*, (December 1994), 37(12), pages 73-84.

[30] Maureen Stone, Ken Fishkin, and Eric Bier. The Movable Filter as a User Interface Tool. In *Human Factors in Computing Systems CHI '94 Conference Proceedings*, pages 306-312.

[31] Ramana Rao, and Stusrt Card. The Table Lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Human Factors in Computing Systems CHI '94 Conference Proceedings*, pages 318-322.

[32] Starbase Graphics Techniques - Volume 1 to 3.